



Armony

accès
libre

Préface de Dominique Burger (BrailleNet)

Accessibilité web

Normes et bonnes pratiques
pour des sites plus accessibles

EYROLLES

Pour que **l'informatique**
soit **un outil**
et non *un ennemi* !

Accessibilité web

Le Web pour tous !

Handicap, déficiences, vieillissement... L'accessibilité est devenue un thème phare, sur le Web comme dans la vie de tous les jours, faisant l'objet de normes souvent incompréhensibles pour le commun des mortels. Ni trop technique, ni trop général, ce livre tord le cou aux idées reçues et donne aux professionnels du Web les clés pour comprendre et réussir une démarche accessibilité.

Après des études de chinois et de science politique, **Armony Altinier** découvre le Web et crée la société ACS Horizons (acs-horizons.fr). Professionnelle engagée, elle accompagne ses clients en tant que consultante et formatrice dans la mise en place d'une communication web plus efficace et plus accessible. Sensibilisée très jeune aux problématiques du handicap, elle milite pour un accès à Internet pour tous sans discrimination.

Également
disponible
au format **DAISY**
Talking Book !



Créez des sites web plus accessibles !

- Comprenez les différents handicaps (visuels, auditifs, moteurs, cognitifs...) et l'usage des technologies d'assistance
- Découvrez le contexte juridique de l'accessibilité, ses normes et la législation en vigueur
- Décryptez le standard WCAG 2 et comprenez les obligations du RGAA (référentiel de l'administration française)
- Évaluez l'accessibilité d'un site selon les 13 thématiques du référentiel AccessiWeb 2.2
- Facilitez l'accès aux contenus (images, multimédia, tableaux, liens, formulaires...) par les lecteurs d'écrans ou la navigation au clavier
- Utilisez avec précaution les couleurs et les scripts
- Apprenez à structurer et présenter vos pages avec HTML/CSS pour faciliter la navigation et la consultation des contenus
- Restez accessible avec HTML5 et JavaScript grâce à WAI-ARIA
- Réussissez votre démarche accessibilité grâce à différentes méthodes (WCAG-EM, Accessibility First Step, MIPAW, AcceDe Web) et selon une logique d'amélioration continue

À qui s'adresse cet ouvrage ?

- Aux développeurs et intégrateurs web qui doivent créer des sites accessibles
- Aux décideurs et responsables de sites Internet qui souhaitent comprendre les enjeux de l'accessibilité
- À tous les métiers impliqués dans la création et la gestion d'un site web : contributeurs, designers web, ergonomes, etc.

www.editions-eyrolles.com

Code éditeur : G12889
ISBN : 978-2-212-12889-5

Accessibilité web

Normes et bonnes pratiques
pour des sites plus accessibles

Ergonomie web. *Pour des sites web efficaces.*

A. BOUCHER.

N°13215, 3^e édition, 2011, 356 pages.

Bien rédiger pour le Web. *Stratégie de contenu pour améliorer son référencement naturel.*

I. CANIVET.

N°12883, 2^e édition, 2011, 552 pages.

Lisibilité des sites web. *Des choix typographiques au design d'information.*

M.-V. BLOND, O. MARCELLIN, M. ZERBIB.

N°12426, 2009, 326 pages.

Réussir son site web avec XHTML et CSS.

M. NEBRA.

N°12485, 3^e édition, 2010, 318 pages.

Réussir un projet de site web.

N. CHU.

N°12742, 6^e édition, 2010, 256 pages.

WordPress 3 pour le blogueur efficace.

F.-X. et L. BOIS.

N°12829, 2010, 320 pages.

Créer son premier thème WordPress pour mobile.

T. BAILLET.

N°13441, 2012, 128 pages.

Réussir son premier site Joomla! 2.5.

H. COCRIAMONT.

N°13425, 2012, 160 pages.

D'OpenOffice.org à LibreOffice 3.5.

S. GAUTIER, G. BIGNEBAT, C. HARDY, M. PINQUIER.

N°13440, 2012, 426 pages.

Premiers pas en CSS et XHTML. *CSS 3 et HTML 5.*

F. DRAILLARD.

N°13388, 4^e édition, 2012, 342 pages.

SPiP 2. *Premiers pas pour créer son site avec Spip 2.0.3.*

A.-L. QUATRAVAUX, D. QUATRAVAUX.

N°12502, 2009, 300 pages.

Préserver son patrimoine numérique.

C. HUC.

N°12789, 2010, 344 pages.

Se protéger sur Internet. *Conseils pour la vie en ligne.*

X. TANNIER.

N°12774, 2010, 232 pages.

Organiser sa veille sur Internet. *Au-delà de Google... Outils et astuces pour le professionnel.*

X. DELENGAIGNE.

N°13295, 2012, 320 pages.

Réussir son blog professionnel. *Image, communication et influence à la portée de tous.*

T. PARISOT.

N°12768, 2^e édition, 2010, 322 pages.

Gimp 2.8. *Débuter en retouche photo et graphisme libre.*

D. ROBERT.

N°12700, à paraître 2013, 380 pages.

Économie du logiciel libre.

F. ELIE.

N°12463, 2009, 195 pages.

Richard Stallman et la révolution du logiciel libre. *Une biographie autorisée.*

R. M. STALLMAN, S. WILLIAMS, C. MASUTTI (Framasoft).

N°12609, 2010, 344 pages.

Design d'expérience utilisateur. *Principes et méthodes.*

S. DAUMAL.

N°13456, 2012, 192 pages.

Card Sorting. *Ne perdez plus vos internautes !*

G. BARRÈRE & É. MAZZONE.

N°13448, 2012, 108 pages.

La stratégie de contenu en pratique. *30 outils passés au crible.*

I. CANIVET & J.-M. HARDY.

N°13510, 2012, 170 pages.

CSS maintenables avec Sass & Compass. *Outils et bonnes pratiques pour l'intégrateur web.*

K. DELOUMEAU-PRIGENT.

N°13417, 2012, 272 pages.

Intégration web : les bonnes pratiques. *Le guide de survie de l'intégrateur !*

C. SCHILLINGER.

N°13370, 2012, 412 pages.

CSS avancées. *Vers HTML 5 et CSS 3.*

R. GOETTER.

N°13405, 2^e édition, 2012, 400 pages.

HTML 5. *Une référence pour le développeur web.*

R. RIMELÉ.

N°12982, 2011, 604 pages.

Ergonomie web illustrée. *60 sites à la loupe.*

A. BOUCHER.

N°12695, 2010, 302 pages (Design & Interface).

Mémento Sites web. *Les bonnes pratiques.*

E. SLOÏM.

N°12802, 3^e édition, 2010, 18 pages.

Mémento Git à 100%.

R. HERTZOG, P. HABOUZIT.

N°13438, 2012, 18 pages.

jQuery 1.7 & jQuery UI. *Une bibliothèque pour JavaScript*

E. SARRION.

N°13504, 2012, 568 pages.

Armony **Altinier**



Accessibilité web

Normes et bonnes pratiques
pour des sites plus accessibles

Préface de Dominique Burger

EYROLLES

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

*Remerciements à Julie Tolomelli (<http://latolo.didrocks.fr>) pour ses illustrations d'ouverture de chapitres,
à Eliza Gapenne pour sa relecture en un temps record,
à Karl Dean (<http://www.ablehere.com>) pour son image de headpointer (chapitre 2),
à Jean-Pierre Villain (Qeios) et à Dominique Burger (BrailleNet) pour leur soutien.*

Préface

Nous sommes tous des utilisateurs du Web, mais nous ne l'utilisons pas toujours, ni tous de la même façon. Tantôt nous y accédons depuis un ordinateur fixe, mais très souvent depuis un ordinateur portable, une tablette ou un smartphone. Certains d'entre nous utilisent des dispositifs capables de traduire en voix de synthèse le texte qui s'affiche à l'écran, ou des appareils à affichage Braille. D'autres, ayant des problèmes pour manipuler le clavier ou la souris, font usage de systèmes à commande vocale, de joysticks particuliers, de claviers virtuels ou d'autres dispositifs adaptés.

Les moyens d'accéder au Web sont donc très divers et nullement limités aux personnes qui lisent parfaitement sur écran et manipulent sans problème la souris et le clavier.

Lorsqu'en 1998, avec des associations de personnes handicapées, des organismes de recherche et des entreprises, nous avons créé l'association BrailleNet, notre objectif était qu'Internet devienne un facteur de réussite dans l'insertion scolaire, universitaire, professionnelle et sociale des personnes handicapées. Dès cet époque, il était clair que le Web pouvait donner à ces personnes accès à des informations et des services qui jusqu'alors leur étaient difficilement accessibles : journaux, livres, encyclopédies, cours en ligne, réservations pour des spectacles ou des voyages, démarches administratives, etc.

Dès son origine, l'un des principes fondateurs du Web a été son universalité, principe posé par son créateur Tim Berners-Lee. Dès 1996, le consortium W3C, chargé de promouvoir et de normaliser Internet au plan mondial, met en place un projet pour veiller à ce que les technologies liées au Web ne constituent pas un obstacle pour les personnes handicapées. C'est la *Web Accessibility Initiative*, ou WAI.

Le travail du W3C et de la WAI a été considérable, puisqu'il a produit des recommandations d'accessibilité couvrant les critères à respecter par les contenus du Web (WCAG¹), par les logiciels de lecture ou les navigateurs web (UAAG²), ainsi que par les logiciels de création de contenus (ATAG³).

Ces standards ont rapidement été reconnus dans la plupart des pays, en Europe notamment⁴, puis par la Commission européenne. La plupart du temps, ils servent de référence aux obligations d'accessibilité des sites web publics⁵.

Cependant, ces standards ne fournissent rien de plus que la référence normative indispensable. Pour que l'accessibilité numérique soit effective, il faut qu'à toutes les étapes du long processus qui produit les applications et les informations constituant un service en ligne, la connaissance et les savoir-faire relatifs à l'accessibilité soient mis en œuvre de manière appropriée.

Prenant conscience que cette compétence humaine était une des clés du succès, l'association BrailleNet a mis en place des formations à destination des professionnels dès 2003. Ces formations ont eu un grand succès, puisque les entreprises doivent de plus en plus souvent justifier d'une connaissance de l'accessibilité dans leurs réponses aux appels d'offres nationaux ou internationaux.

L'association BrailleNet a cherché à faire de ces formations le creuset d'une compétence collective, en créant et en animant une communauté professionnelle, le groupe de travail AccessiWeb. Ce groupe, qui compte aujourd'hui plus de 450 membres, contribue par ses échanges et ses discussions à faire évoluer la compréhension en profondeur des standards d'accessibilité du W3C, de leur raison d'être, mais aussi des techniques et des méthodes qui les sous-tendent et permettent de les mettre en œuvre.

Dans ce contexte, il manquait un ouvrage de synthèse faisant un état de la question, utilisable bien au-delà de cette communauté, par les professionnels du Web concernés par l'accessibilité (ils devraient tous l'être !). Cette lacune est désormais réparée avec ce livre d'Armony Altinier, dont je salue la clarté et le talent pédagogique. Cet ouvrage solide repose sur une longue pratique de l'auteure dans le domaine de l'accessibilité du Web, depuis plusieurs années. Je me réjouis également

1. *Web Content Accessibility Guidelines*

2. *User Agent Accessibility Guidelines*

3. *Authoring Tool Accessibility Guidelines*

4. La première circulaire française demandant leur application, émise par le Premier Ministre, date de 1999.

5. En France, la loi n° 2005-102 du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées, et son décret d'application n° 2009-546 du 14 mai 2009 créant un référentiel d'accessibilité des services de communication publique en ligne, se réfèrent à ces standards pour définir l'obligation concernant l'accessibilité des services de communication en ligne de l'État, des collectivités territoriales et des établissements publics qui en dépendent.

du fait qu'elle ait associé à ce travail les meilleurs experts français du domaine, pour produire un ouvrage complet qui devrait être un outil de travail aussi bien pour les développeurs que pour les responsables de projets Web.

Dominique Burger

Institut National de la Santé et de la Recherche Médicale
Université Pierre et Marie Curie
Président de l'association BrailleNet

Table des matières

Avant-propos	1
---------------------------	----------

PREMIÈRE PARTIE

Accessibilité, handicap et Web	7
---	----------

CHAPITRE 1

L'accessibilité du Web, qu'est-ce que c'est ?	9
--	----------

Accessibilité et handicap	10
---------------------------------	----

Le Web, technologie de l'universel	12
--	----

Naissance de l'accessibilité du Web	13
---	----

CHAPITRE 2

L'accessibilité : pour qui ? Pourquoi ?	15
--	-----------

Les différents types de handicaps	17
---	----

Combien de personnes concernées ?	17
---	----

Quels types de limitations ?	18
------------------------------------	----

<i>Handicaps visuels</i>	18
--------------------------------	----

<i>Handicaps auditifs</i>	20
---------------------------------	----

<i>Handicaps moteurs</i>	21
--------------------------------	----

<i>Handicaps « mentaux »</i>	22
------------------------------------	----

<i>Troubles de santé invalidants</i>	25
--	----

Handicap et vieillissement	25
----------------------------------	----

Les technologies d'assistance	26
-------------------------------------	----

Logiciels de lecture d'écran et synthèse vocale	28
---	----

Logiciels de lecture d'écran et plage Braille	30
---	----

Logiciels de grossissement	31
----------------------------------	----

<i>Headstick et pointeurs laser</i>	33
---	----

Logiciels de saisie de texte et claviers virtuels	33
---	----

Et le handicap auditif alors ?	35
--------------------------------------	----

Des besoins spécifiques, comme tout le monde !	35
--	----

La tentation du test utilisateur en guise d'évaluation de l'accessibilité	37
---	----

CHAPITRE 3

Contexte technologique et juridique 39

Évolutions du Web et des technologies d'assistance	40
Vers un Web « sémantique » ?	40
Les défis du Web riche pour l'accessibilité	41
Accessibilité web et obligation légale	43
Le développement de l'accessibilité par la loi	43
Le contexte législatif français	44
<i>La loi pour l'égalité des droits et des chances, la participation et la citoyenneté des per-</i>	
<i>sonnes handicapées</i>	<i>44</i>
<i>Le décret d'application.</i>	<i>46</i>
<i>L'arrêté ministériel relatif au référentiel général d'accessibilité pour les</i>	
<i>administrations : que faut-il faire concrètement ?</i>	<i>46</i>
<i>Quelle obligation d'accessibilité en dehors du secteur public ?</i>	<i>53</i>

DEUXIÈME PARTIE

Aspects techniques de l'accessibilité du Web 55

CHAPITRE 4

Comprendre les règles d'accessibilité du Web..... 57

Un seul standard, les WCAG	58
La création du groupe de travail WAI par le W3C	59
Les quatre grands principes des WCAG 2	61
Douze grandes règles issues des quatre principes	61
Les critères de succès	63
Les techniques et échecs WCAG	64
Les trois niveaux de conformité : A, AA et AAA	67
Évaluer l'accessibilité d'un site web	70
Principe de l'audit	70
Méthodes d'évaluation	72
<i>Le référentiel AccessiWeb.</i>	<i>73</i>
<i>Le RGAA</i>	<i>77</i>
Différences entre le référentiel AccessiWeb et le RGAA	80
<i>Rédaction, méthodologie et structure</i>	<i>80</i>
<i>Maintenance et évolutions</i>	<i>81</i>
<i>Contenus.</i>	<i>81</i>

CHAPITRE 5

Introduction au référentiel AccessiWeb 85

CHAPITRE 6

Images, cadres et couleurs (thématiques AccessiWeb 1 à 3)..... 89

Images	89
Un attribut alt sur chaque image	90
Images porteuses d'information, images de décoration et images d'illustration	91
Rédiger une alternative pertinente	96
<i>Description courte</i>	96
<i>Description longue</i>	97
<i>Cas particuliers : captchas et images tests</i>	99
Les textes sous forme d'image	103
<i>Cas particuliers</i>	105
Cadres	105
Un attribut title pertinent	105
Copier-coller = danger !	107
Couleurs	107
Alternative aux informations données par la couleur	108
<i>Pertinence de l'alternative</i>	111
Un contraste « suffisant »	113
<i>Les limites de la seule prise en compte des contrastes</i>	115

CHAPITRE 7

Multimédia (thématique AccessiWeb 4)..... 119

Vocabulaire : média temporel, média non temporel, synchronisé ou non, préenregistré ou en direct	120
Média temporel versus média non temporel	120
Média synchronisé ou non synchronisé	121
Média préenregistré ou en direct	121
Transcription et sous-titrage : écrire ce qui ne peut être entendu	122
Et les sourds qui ne savent pas lire ?	124
Audiodescription : dire ce qui ne peut être vu	126
Un lecteur accessible	127
Description synthétique du média	127
Un média contrôlable par l'utilisateur	129
Un média « compatible avec les technologies d'assistance »	130
Médias de décoration et autres cas particuliers	131

CHAPITRE 8

Tableaux (thématique AccessiWeb 5) 133

Tableaux de données et tableaux de présentation	134
Les tableaux de présentation	134
Les tableaux de données	136
Réaliser un tableau de données accessible	137
Exemple d'un tableau simple	137
Exemple d'un tableau simple à double entrées	141
Exemple d'un tableau complexe à en-têtes multiples	146
Un tableau accessible, facile ?	155

CHAPITRE 9

Liens et scripts (thématiques AccessiWeb 6 et 7) 157

Liens	157
Intitulé et titre de lien	158
Un intitulé présent et explicite	158
<i>L'ambiguïté du terme « explicite »</i>	159
<i>Contexte de lien</i>	159
<i>Le cas des images liens</i>	162
<i>Les liens composites</i>	163
Halte aux attributs title abusifs !	165
Le cas des liens identiques	165
Scripts	166
Alternative à JavaScript	167
<i>Une alternative « si nécessaire »</i>	172
<i>Le cas particulier de l'environnement maîtrisé</i>	173
Un script accessible	174
<i>Navigation au clavier et à la souris</i>	174
<i>Des scripts compatibles avec les technologies d'assistance</i>	175
Les changements de contexte	176
<i>Une définition complexe</i>	176
<i>Rendre un changement de contexte accessible</i>	177
Contrôler les alertes non sollicitées	181

CHAPITRE 10

Éléments obligatoires, structuration de l'information et présentation de l'information (thématiques Accessiweb 8 à 10) 183

Éléments obligatoires	183
Sécuriser l'accès des technologies d'assistance à la page web	184
<i>Définir un cadre technique clair</i>	184

<i>Ne pas induire en erreur les agents utilisateurs</i>	190
Indiquer les modifications au sein de la page	191
<i>Les changements de langue au sein de la page</i>	191
<i>Les changements de sens de lecture</i>	194
Structuration de l'information	195
Des titres pour structurer la page	196
Des listes correctement balisées	199
Des abréviations et acronymes explicités	202
Des citations déclarées comme telles	204
Présentation de l'information	207
Utiliser les feuilles de styles	208
Rendre l'information perceptible	210
<i>Désactiver les images ou les CSS</i>	210
<i>Le texte est-il lisible avec un zoom texte à 200 % ?</i>	216
<i>Des textes cachés restitués si nécessaire</i>	219
Se repérer dans la page	220
Lisibilité sur écran et dyslexie	225
Ne pas donner l'information seulement par la forme, la taille ou la position ..	227

CHAPITRE 11

Formulaires, navigation et consultation

(thématiques AccessiWeb 11 à 13)..... 229

Formulaires	229
Étiquettes et boutons : des intitulés pertinents	230
<i>Des étiquettes présentes, pertinentes et associées aux champs</i>	230
<i>Des boutons à l'intitulé pertinent</i>	236
Structurer ses formulaires	240
<i>Regrouper les informations de même nature</i>	240
<i>Structurer les listes de choix</i>	242
Encadrer la saisie des données	245
<i>Accompagner la saisie des données</i>	245
<i>Contrôler le traitement des données</i>	250
Navigation	251
Les systèmes de navigation	252
<i>Dans un ensemble de pages : navigation principale, plan du site, recherche</i> ..	252
<i>Dans une collection de pages : liens suivant/précédent</i>	252
<i>Optimiser sa navigation : le fil d'Ariane</i>	252
Les zones de navigation : liens d'accès rapide et liens d'évitement	253
Attention, c'est un piège !	255
Consultation	256
Maîtriser son environnement : les limites de temps et l'ouverture de nouvelles fenêtres	257

<i>Rafraîchissement ou redirection automatique et limites de temps</i>	258
<i>Ouverture de nouvelles fenêtres</i>	258
Documents en téléchargement	262
<i>Identifier les documents en téléchargement</i>	262
<i>Des documents bureautiques accessibles</i>	264
Des textes clairs et explicites	266
<i>Attention au jargon !</i>	266
<i>Contenus cryptiques (émoticônes, art ASCII)</i>	266
<i>Prononciation</i>	267
Effet de flash, contenus en mouvement et clignotants	268

CHAPITRE 12

HTML5, ARIA et accessibilité..... 271

Chouette, encore plus de sémantique !	274
Une meilleure structuration de la page	274
Nouvel algorithme de titrage	277
Des légendes pour les images	278
Des champs de formulaire mieux identifiés	279
Le multimédia devient du Web	279
L'interactivité définie en HTML	281
WAI-ARIA, le Web riche accessible	282
Comment ça marche ?	283
<i>Les rôles, avec l'attribut role</i>	284
<i>Propriétés et états</i>	292
WAI-ARIA, « la » solution ?	299
Et maintenant, que faire ?	302

TROISIÈME PARTIE

L'accessibilité du Web, au-delà des normes 305

CHAPITRE 13

La démarche accessibilité 307

Par où commencer ?	308
Deux axes à concilier : ce qui est faisable facilement et ce qui est crucial pour l'accessibilité	308
<i>WCAG-EM : la méthodologie selon la WAI</i>	309
<i>Accessibility First Step : l'étape juste avant de faire de l'accessibilité</i>	310
<i>MIPAW : l'accès à l'information d'abord</i>	311
<i>AcceDe Web : l'accessibilité selon son profil et selon la phase projet</i>	312
« Et l'accessibilité ? » : une démarche en une question	312

Une démarche progressive et continue : accessibilité et conduite du changement . .	315
Se former, sensibiliser, convaincre	315
Une amélioration continue : « tendre vers » l'accessibilité	316
L'accessibilité... accessible à tous ?	317
Les métiers de l'accessibilité du Web : de la formation à la professionnalisation . . .	317
<i>Spécialisation et transversalité</i>	318
<i>Les métiers dédiés à l'accessibilité : évaluateur, consultant, formateur</i>	318
<i>La spécialisation nécessaire des intégrateurs et des contributeurs</i>	319
Tout le monde peut contribuer !	320
Conclusion : Accessibilité et liberté	321
Définition des abréviations par ordre alphabétique	325
Index	327

Avant-propos

- Bonjour, je cherche des livres à recommander en vue d'une formation que je vais donner sur l'accessibilité du Web. Je n'en ai pas trouvé, avez-vous prévu d'en publier ?
- Vous cherchez à acheter un livre ? Ne voulez-vous pas plutôt l'écrire ?
- Euh... Ben non, pas vraiment...

Après une bonne heure d'échange, les talents de négociatrice de Muriel Shan Sei Fan, responsable du secteur informatique aux éditions Eyrolles, avaient fini par me convaincre de relever ce défi : écrire un livre sur l'accessibilité du Web.

Et quel défi ?! L'accessibilité du Web est un domaine presque aussi ancien que le Web lui-même. Cette préoccupation de l'accès à Internet pour tous a d'ailleurs été posée dès l'origine comme un des objectifs fondamentaux du W3C (*World Wide Web Consortium*), consortium international en charge de l'édiction des standards du Web.

Pourtant, cette matière demeure un mystère pour beaucoup.

« Mais qu'entends-tu exactement par accessibilité du Web ? Peux-tu juste me dire par e-mail, en deux mots, comment faire pour rendre mon site accessible ? »

En deux mots, par e-mail... non, désolée. Je ne sais pas faire et ne veux pas faire. Et pourtant, combien de fois n'ai-je entendu cette question, au demeurant légitime et qui a le mérite de prouver l'intérêt de la personne pour le sujet ?

Mon refus de faire de l'accessibilité de comptoir, par courriel ou messagerie instantanée en deux minutes, peut être mal perçu. Or, donner deux ou trois astuces pour rendre un site plus accessible sans prendre le temps de comprendre de quoi il s'agit

reviendrait à transformer une question de société en question purement technique, risquant ainsi de passer totalement à côté du sujet et d'induire mon interlocuteur en erreur. Ce type de démarche me semble dangereux et contre-productif pour une accessibilité du Web sur le long terme.

On peut définir l'accessibilité du Web par le fait de permettre à tous d'accéder à Internet, y compris des personnes devant utiliser des technologies particulières en raison d'un handicap. De cette définition découlent des règles permettant une application technique de l'accessibilité. Mais si l'on ne comprend pas les conséquences d'une navigation web en situation de handicap, on risque de mal interpréter les règles.

Prenons un exemple concret : l'alternative textuelle aux éléments visuels. Cette règle, simple en apparence, vise à donner aux personnes aveugles et malvoyantes accès aux contenus visuels via un petit texte décrivant l'image. Malheureusement, se contenter de cette définition sans prendre le temps d'en comprendre l'objectif conduit très souvent à une mauvaise utilisation de ces textes alternatifs qui, s'ils ne sont pas pertinents, viennent encore alourdir la navigation et constituent un frein supplémentaire à l'accessibilité de la page. On peut ainsi rendre encore plus inaccessible un contenu en pensant bien faire.

« Euh... attends une minute... Tu veux dire qu'un aveugle peut surfer sur le Web ?! »

Oui, mais ce n'est qu'un exemple. L'accessibilité du Web *s'adresse à tous*, pas uniquement aux personnes ayant un handicap visuel, comme nous allons le voir en détail.

Paradoxalement, l'accessibilité du Web semble peu accessible, alors qu'elle concerne tout le monde. Entre débats d'experts et questions de néophytes, l'écart est souvent grand, et c'est la question à laquelle tentera de répondre ce livre : comment (ré)concilier deux univers, l'accessibilité et le Web, au profit des premiers concernés – les personnes handicapées ?

En effet, on ne peut parler d'accessibilité sans parler de handicap. Comprendre les besoins spécifiques des personnes en situation de handicap et leur façon de naviguer sur le Web est un prérequis indispensable pour faire de l'accessibilité de qualité. C'est ce que nous verrons dans **la première partie** de cet ouvrage.

Une fois que l'on a une vision globale de la variété et de la complexité des usages à couvrir, se pose alors la question de la mise en œuvre technique de l'accessibilité du Web, ce qui passe par le respect de règles et l'adoption d'une méthodologie d'évaluation. Cela fait l'objet de **la deuxième partie** de cet ouvrage.

Enfin, et parce qu'il ne suffit pas de respecter des règles à un temps *t* pour faire de l'accessibilité, nous verrons également les autres dimensions de cette dernière, tant sur la pratique que l'on peut avoir du Web que sur la démarche à mettre en œuvre

pour un maintien de l'accessibilité du Web dans le temps. Car plus que l'application de règles techniques par un expert, la démarche accessibilité requiert un engagement de toute la chaîne de production web. Nous abordons cet aspect dans la **troisième partie** de cet ouvrage.

« Règles d'accessibilité, méthodologie d'évaluation, chaîne de production web... Ça m'a l'air bien compliqué tout ça... Ce livre est-il vraiment fait pour moi ?... »

Eh bien, ça dépend. Si vous lisez ces pages, c'est qu'à priori, le sujet vous intéresse ou du moins vous intrigue, ce qui est un bon début.

Ce livre ne parlera que d'accessibilité du Web, sujet suffisamment large pour nous occuper tout un ouvrage. Autrement dit, il n'est pas question ici de réapprendre le HTML ou les CSS, ni même de parler d'ergonomie.

RÉFÉRENCES HTML, CSS, ergonomie...

Pour les développeurs soucieux d'améliorer leurs connaissances du Web, nous vous recommandons la lecture des livres suivants :

- 📖 Mathieu Nebra, *Réussir son site web avec XHTML et CSS*, 3^e édition, Eyrolles, 2010
- 📖 Raphaël Goetter, *CSS 2 : Pratique du design web*, 3^e édition, Eyrolles, 2009
- 📖 Raphaël Goetter, *CSS avancées*, 2^e édition, Eyrolles, 2012
- 📖 Rodolphe Rimelé, *HTML 5 : une référence pour le développeur web*, Eyrolles, 2011
- 📖 Corinne Schillinger, *Intégration web : les bonnes pratiques*, Eyrolles, 2012
- 📖 Amélie Boucher, *Ergonomie Web : Pour des sites web efficaces*, 3^e édition, Eyrolles, 2011

Vous pouvez par ailleurs assister à Paris Web, qui propose des conférences au mois d'octobre à Paris sur la qualité web sous toutes ses formes et faisant une large place à l'accessibilité.

Pour visionner les vidéos des précédentes conférences, consultez le site Paris Web :

► <http://www.paris-web.fr>

Les première et troisième parties de ce livre peuvent être lues par toute personne sachant utiliser Internet, sans autre prérequis technique. La deuxième partie, en revanche, s'adresse à un public un peu plus averti, pour lequel la vue de quelques lignes de code HTML est accueillie avec curiosité ou enthousiasme. Ainsi, si les mots « balise » ou « attribut » vous donne la migraine, rien ne vous empêche de sauter les passages plus techniques tout en profitant du reste du livre.

L'objectif de ce livre n'est pas de former des experts, mais bien de s'adresser au plus grand nombre pour faire découvrir en quoi consiste l'accessibilité du Web. Si rien ne remplace la formation et l'expérience, il a paru utile de réunir les connaissances éparses sur le sujet en un ouvrage structuré et cohérent, afin d'offrir une vision d'ensemble de l'accessibilité du Web et de donner envie d'aller plus loin.

Car telle est l'ambition de ce livre : partager avec vous la magie d'un Web auquel chacun peut accéder, et vous montrer qu'on peut tous y contribuer.

Mais trêve de bavardage ! Entrons dans le vif du sujet. Vous me suivez ?

À SAVOIR Un livre accessible à tous !

Ce livre est également disponible en téléchargement au format Daisy Talking Book, pour permettre aux personnes aveugles et malvoyantes d'accéder à son contenu sous forme audio.

Attention à bien paramétrer votre lecteur d'écran, si vous êtes aveugle, pour une prononciation correcte des abréviations et acronymes. Un glossaire est prévu en annexe pour vous y aider.

Par ailleurs, à la demande de l'auteure et en accord avec les éditions Eyrolles, le contenu du livre sera librement téléchargeable, sous licence Creative Commons BY-NC-ND, après 1 200 exemplaires vendus, afin que chacun puisse s'initier librement à l'accessibilité du Web.

Remerciements

L'écriture de ce livre s'est faite dans un contexte personnel difficile, très en rapport avec le sujet, entre traitements, aliments, opération... Mes premiers remerciements vont donc à mon mari, Nicolas Didon, sans l'appui duquel je n'aurais sans doute pas été au bout.

Je tiens également à remercier très chaleureusement Jean-Pierre Villain. Vous verrez son nom apparaître dans ce livre en tant qu'acteur important de l'accessibilité du Web en France. Il a accepté de relire, mais bien au-delà, ses encouragements, ses conseils et le temps qu'il a accepté de consacrer à nos échanges passionnés ont réellement appuyé ma motivation dans la dernière ligne droite de l'écriture, la phase la plus difficile, quand on croit ne plus voir la fin. Merci Jean-Pierre !

Des remerciements également pour mes éditrices esclavagistes préférées (je dois être masochiste...) : Muriel Shan Sei Fan et Karine Joly. Pour m'avoir fait confiance et avoir fait preuve de patience quand les délais ont dû être allongés. Vous voyez, les filles, ça y est, vous l'avez sorti votre livre sur l'accessibilité web !

Merci encore à Dominique Burger pour la préface, à Julie Tolomelli pour ses superbes illustrations de chapitre et à tous mes relecteurs, par ordre alphabétique : Dominique Burger, Nicolas Didon, Matthieu Faure, Rodolphe Rimelé (pour la partie sur HTML5), Christelle Thomas et Jean-Pierre Villain.

Enfin, je dédie ce livre à mon Papa, Patrick Altinier, qui n'aura malheureusement pas pu le voir terminé, mais qui était déjà très fier en sachant que je l'écrivais.



PREMIÈRE PARTIE

Accessibilité, handicap et Web

À l'époque où, pour des raisons de santé, j'ai dû garder la chambre pendant plusieurs mois, Internet m'est soudain apparu comme une fenêtre sur le monde absolument indispensable. Quand sortir est impossible et que le moindre acte de la vie quotidienne se transforme en parcours du combattant, le Web devient un espace vital, tant pour travailler que pour se distraire, faire ses courses ou garder contact avec ses proches.

Or, un jour, alors que je me lève et m'apprête à me rendre sur l'ordinateur, une violente douleur m'empêche d'utiliser ma main droite. Qu'à cela ne tienne, je décide d'utiliser ma main gauche. Mais se servir d'une souris de la main gauche quand on est droitier devient vite très pénible. J'opte alors pour la navigation au clavier. À l'aide de la touche *Tabulation*, j'essaie donc de me rendre sur un site d'information que je consulte régulièrement. Cette expérience s'avère longue et laborieuse, jusqu'à ce que je me rende compte, étonnée, que le lien que je souhaite atteindre est inaccessible au clavier ! Étrange... Comment est-ce possible, hier encore je pouvais cliquer dessus avec ma souris ?... La réponse est que le site n'a pas été pensé pour être accessible.

En effet, à l'instar de tout espace public, les sites web peuvent être conçus de façon accessible ou non. Car qu'est-ce que le Web sinon un espace numérique sur lequel on peut naviguer grâce à des liens reliant les pages entre elles ? Dans le monde physique, les lieux peuvent être plus ou moins accessibles, selon que l'architecte ou le client a pris en compte l'accessibilité dans ses critères de conception. C'est exactement la même chose dans le monde numérique : *l'accessibilité tient d'abord et avant tout au fait de l'avoir prévue en amont.*

Tel est l'objet de cette première partie : dresser un panorama de l'accessibilité du Web – qu'entend-on exactement par « accessible à tous » (chapitre 1), quels sont les utilisateurs de sites accessibles et quels sont leurs besoins (chapitre 2) et sur quelles normes peut-on s'appuyer pour bâtir des sites accessibles (chapitre 3) ? Une introduction aux règles techniques d'accessibilité suivra dans la deuxième partie.

L'accessibilité du Web, qu'est-ce que c'est ?

« L'accessibilité du Web signifie que les personnes en situation de handicap peuvent utiliser le Web. Plus précisément, qu'elles peuvent percevoir, comprendre, naviguer et interagir avec le Web, et qu'elles peuvent contribuer sur le Web. L'accessibilité du Web bénéficie aussi à d'autres, notamment les personnes âgées dont les capacités changent avec l'âge.

L'accessibilité du Web comprend tous les handicaps qui affectent l'accès au Web, ce qui inclut les handicaps visuels, auditifs, physiques, de parole, cognitifs et neurologiques. »

Définition de l'initiative pour l'accessibilité du Web (WAI, Web Accessibility Initiative), groupe de travail du W3C.



Prévoir un espace accessible à tous pose plusieurs questions. Tout d'abord, il convient de déterminer les personnes auxquelles cette accessibilité s'adresse. Qu'entend-t-on exactement par « accessible à tous » ?

Accessibilité et handicap

Définir l'accessibilité du Web est un enjeu en soi, tant la compréhension du sujet peut varier selon les personnes. Or, le Web n'est finalement qu'une partie du réel appliqué à l'informatique, et l'ambiguïté dont est porteuse la notion d'accessibilité du Web se retrouve quand on parle d'accessibilité dans le monde physique.

Le mot « accessibilité » véhicule deux sens complémentaires : d'une part la possibilité d'accéder, et d'autre part l'égal accès. Rien n'indique dans ce terme que l'on s'adresse spécifiquement aux personnes handicapées, tout le monde pouvant être concerné par des questions d'accès à un lieu ou un contenu. Il est pourtant important de comprendre l'accessibilité du Web comme étant tout d'abord un enjeu pour les personnes handicapées, bien que les résultats bénéficient à tous.

Le choix des mots n'est pas anodin et résulte d'une histoire complexe et de l'évolution du rapport qu'entretient la société vis-à-vis du handicap.

Encore au XX^e siècle, on ne parlait pas d'accessibilité. L'approche médicale qui dominait alors considérait que la personne « déficiente » avait un problème. C'était donc à elle de s'adapter au monde qui l'entourait. À défaut d'y parvenir, on parlait alors de personnes « inadaptées ».

Des événements historiques bousculèrent petit à petit cette approche médicale. La révolution industrielle et les deux guerres mondiales, avec leur cortège de personnes mutilées, déplacèrent la question de l'infirmité du champ médical à la sphère sociale. La société n'avait-elle pas en effet une responsabilité envers ces familles dont l'un des membres s'était sacrifié pour le pays ? Le nombre de victimes imposa une réponse sociale et politique. Refusant d'être reléguées au rang de citoyens de seconde zone, les personnes concernées s'organisèrent en collectifs pour que les malades et infirmes ne soient plus « assistés » mais que leur droit à accéder à une vie normale leur soit donné, au même titre que pour tous. On passe ainsi du registre de la charité à celui du droit.

Le problème de l'inadaptation se déplaça donc progressivement de la personne elle-même vers l'environnement. Ce n'est plus aux personnes de s'adapter, mais à leur environnement d'être accessible.

Cette nouvelle terminologie s'imposa petit à petit, associant handicap et accessibilité, tout en contextualisant la problématique au point que toute personne peut aujourd'hui être considérée « en situation de handicap », même temporairement.

VOCABULAIRE « Hand in cap »

Il est intéressant de se pencher sur l’étymologie du mot « handicap » pour s’apercevoir qu’il n’a pris son sens actuel qu’au début du XX^e siècle. Selon le *Oxford English Dictionary*, c’est en 1653 qu’on recense l’expression « hand in cap » pour la première fois, en référence à un jeu de hasard permettant à deux personnes d’estimer la valeur d’objets qu’ils voulaient échanger. Cela se faisait de façon confidentielle, en mettant la main dans un chapeau, avec pour arbitre une tierce personne qui déterminait la valeur de chaque objet.

Puis, cette expression est passée dans le domaine des courses hippiques. Là encore, la notion de hasard prédomine. Pour permettre aux parieurs de faire leur choix en toute équité, on place un handicap sur les concurrents les plus rapides pour égaliser les chances. La course à handicap existe toujours. À noter que dans le cas du golf, le handicap est au contraire un avantage donné aux concurrents les moins habiles, mais toujours pour équilibrer les chances.

Enfin, le handicap a d’abord été étendu de façon métaphorique au champ économique, avant d’être utilisé pour décrire des personnes atteintes de déficience.

UN PEU D’HISTOIRE Le rapport Bloch-Lainé

En 1967, un haut fonctionnaire, François Bloch-Lainé, remet un rapport au Premier ministre de l’époque, Georges Pompidou, intitulé « Étude du problème général de l’inadaptation des personnes handicapées ». Tout un programme ! L’intitulé du rapport illustre bien la façon dont la question était envisagée à l’époque... Ce texte ouvrira la voie au texte d’orientation de 1975.

Pour une chronologie de la politique du handicap, consultez le site suivant :

► <http://ur1.ca/9vo2b^a>

a. <http://www.vie-publique.fr/politiques-publiques/politique-handicap/chronologie/>

À SAVOIR Une tendance mondiale en faveur de l’accessibilité

Le changement sémantique évoqué précédemment s’inscrit dans le cadre d’un élan international et européen. C’est d’ailleurs suite à un effort européen pour encourager tous les États membres à promouvoir l’accessibilité numérique que la France a voté la loi Handicap de 2005.

Voir, par exemple, la communication de la Commission au Conseil, au Parlement européen, au Comité économique et social et au Comité des régions intitulée « eEurope 2005 : une société de l’information pour tous » :

► <http://ur1.ca/a402d^a>

Ce texte indique notamment que « l’inclusion numérique représente un des thèmes horizontaux communs à tous les domaines du plan d’action eEurope 2005. Une attention particulière doit être accordée à l’établissement de normes européennes en matière d’accessibilité au réseau, des lignes directrices sur l’initiative d’accessibilité du Web (WAI) et des règles pour un étiquetage commun des pages web accessibles. »

a. http://europa.eu/legislation_summaries/information_society/strategies/l24226_fr.htm

La loi française du 11 février 2005 a entériné partiellement ce glissement terminologique, en donnant la définition suivante du handicap :

« Constitue un handicap, au sens de la présente loi, toute limitation d'activité ou restriction de participation à la vie en société *subie dans son environnement* par une personne en raison d'une altération substantielle, durable ou définitive d'une ou plusieurs fonctions physiques, sensorielles, mentales, cognitives ou psychiques, d'un polyhandicap ou d'un trouble de santé invalidant. »

Or, si cette définition du handicap est relativement tardive en droit français, elle s'inscrit dans un mouvement international plus large. Car cette transformation de l'approche du handicap est globale et porte en elle une dimension d'universalité, qu'une révolution technologique ne tardera pas à mettre en valeur.

Le Web, technologie de l'universel

Inventé dans les années 1960, Internet est un réseau de réseaux permettant l'échange de données de différentes natures et sous divers protocoles. On peut citer le courrier électronique, mais aussi aujourd'hui la voix sur IP, les flux vidéo ou encore le Web bien entendu.

Le Web n'est qu'un des éléments qu'on peut trouver sur Internet, et c'est pourtant sa facilité d'utilisation, reposant sur des pages reliées entre elles par des liens hypertextes, qui donna la dimension quasi universelle qu'on lui connaît aujourd'hui. Or, cette universalité souhaitée dès l'origine en 1989 par son inventeur, Tim Berners-Lee, n'est possible qu'à condition que chaque développeur de site utilise un même langage, de façon standardisée, afin de rendre possible l'affichage du contenu du site chez tous les internautes, quelle que soit leur façon de consulter la page, et à condition d'utiliser un navigateur respectant lui aussi les standards.

C'est justement l'objectif premier que se donna la toute nouvelle structure créée en octobre 1994 par Tim Berners-Lee pour standardiser les langages du Web, le *World Wide Web Consortium*, abrégé en W3C. Il définit sa mission comme devant conduire le Web à son potentiel maximal en développant des protocoles et règles permettant d'assurer le développement du Web sur le long terme.

La mission du W3C est donc d'assurer un « Web pour tous », ce qui est précisé de la façon suivante sur leur site :

« La valeur sociale du Web vient du fait qu'elle permet aux Hommes de communiquer, commercer et offre des opportunités pour partager leurs connaissances. L'un des objectifs premiers du W3C est de rendre ces avantages disponibles pour tous, quels que soient leur matériel, logiciel,

infrastructure réseau, langue maternelle, culture, localisation géographique, *ou leurs aptitudes physiques ou mentales.* »

SOURCE Le Web pour tous du W3C

Voir la citation originale en anglais à l’adresse suivante :

► <http://ur1.ca/6dfqh^a>

a. <http://www.w3.org/Consortium/mission>

Quand on réfléchit deux secondes à cette première mission, on ne peut que dire : « Wouahou ! Quelle ambition ! Un Web pour tous, quelle que soit sa langue maternelle ? Un peu comme le poisson de Babel imaginé par Douglas Adam dans ses romans qui permet, quand on le glisse dans l’oreille, d’avoir une traduction de toutes les langues de la galaxie en temps réel ?! » Que dire alors de la localisation géographique, avec les problèmes de censure dans certains pays, ou de l’infrastructure réseau, quand aujourd’hui encore, dans certaines petites communes de France, l’ADSL n’est pas disponible ?!

Bref, vaste chantier, magnifique utopie diront certains, et pourtant, une utopie en marche. Il suffit de repenser à ce qu’étaient nos vies avant le Web pour voir l’immense révolution qu’il constitue.

Cette volonté d’universalité est clairement affichée par le W3C, et prend en compte dès le départ la question du handicap. Mais comme on vient de le dire, cette volonté d’universalité est tellement globale qu’il est impossible de travailler sur chacun de ces aspects en même temps. C’est pourquoi plusieurs chantiers ont été ouverts pour creuser plus précisément chaque question, dont l’un consacré spécifiquement à l’accessibilité du Web.

Naissance de l’accessibilité du Web

Imaginé à l’automne 1996 par quelques membres de l’équipe du W3C, un groupe de travail autonome dédié à l’accessibilité du Web se mit en place pour travailler spécifiquement sur cette question. Le 7 avril 1997, l’initiative pour l’accessibilité du Web, appelée aussi la WAI (*Web Accessibility Initiative*), était officiellement lancée par le W3C.

VOCABULAIRE A11y

Une abréviation courante que vous rencontrerez pour signifier l’accessibilité du Web est « a11y ». Elle est composée du « A » initial et du « Y » final du mot anglais *accessibility*, et les onze lettres du milieu sont remplacées par le chiffre 11.

Cette abréviation est inspirée des abréviations i18n pour *internationalization* et l10n pour *localization*.

Bien que s'inscrivant dans la mission d'universalité du W3C, la définition de l'accessibilité du Web donnée par la WAI est spécifiquement dédiée à l'accès au Web par les personnes handicapées :

« *L'accessibilité du Web signifie que les personnes handicapées peuvent utiliser le Web. Plus précisément, l'accessibilité du Web signifie que les personnes handicapées peuvent percevoir, comprendre, naviguer et interagir avec le Web, et qu'elles peuvent contribuer sur le Web. L'accessibilité du Web bénéficie aussi à d'autres, notamment les personnes âgées dont les capacités changent avec l'âge.* »

SOURCE L'accessibilité du Web selon la WAI

Voir la citation originale en anglais sur le site de la WAI :

► <http://ur1.ca/9vo3x^a>

a. <http://www.w3.org/WAI/intro/accessibility.php>

C'est cette définition de l'accessibilité du Web donnée par la WAI, la seule qui soit réellement dédiée à ce domaine, qui servira de fil conducteur tout au long de ce livre.

Pas question ici de parler de multilinguisme, de respect des standards ou des autres considérations mentionnées par l'objectif que s'est donné le W3C, mais bien de l'accès au Web par les personnes handicapées. Cette précision est importante tant la confusion règne dans le monde du Web quand on parle d'accessibilité. Mais ce champ restreint n'en est pas moins vaste et nous verrons qu'en tant qu'élément essentiel de la qualité web, l'accessibilité a un impact très fort sur les autres domaines du Web.

CONCEPT Universal Design

La WAI s'inscrit explicitement dans le mouvement plus large de la conception universelle ou *Universal Design*, encore appelé *Design for All*. Ce mouvement mondial, né dans les années 1970-1980, est tout d'abord l'œuvre d'architectes souhaitant promouvoir le design comme vecteur d'émancipation et de participation pour tous. L'idée générale est de ne pas penser des espaces spécialement adaptés pour certaines personnes - handicapées, par exemple - à l'écart de l'espace général, mais de prévoir un espace accessible à tous dès la conception.

La vision de la WAI va dans le même sens. Un site web accessible ne doit pas être différent ou à l'écart du site principal, mais c'est le site principal lui-même qui doit être pensé pour être accessible dès le départ. Les sites comportant des liens vers une « version accessible » ne sont donc pas accessibles.

Voir l'argumentaire en anglais de la WAI intitulé « L'accessibilité contribue à l'Universal Design » :

► <http://ur1.ca/9vo4g^a>

a. <http://www.w3.org/Talks/WAI-Intro/slide8-0.html>

Mais avant de voir cela, commençons par découvrir qui sont ces personnes handicapées à qui la WAI s'adresse, quelles sont leurs particularités et leurs besoins sur le Web.

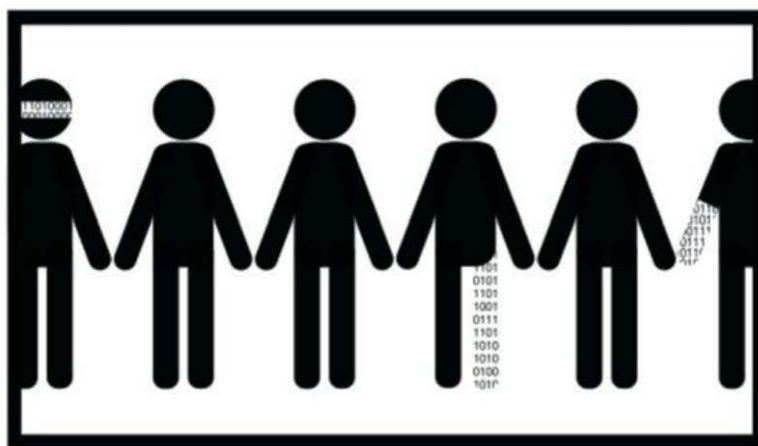
L'accessibilité : pour qui ? Pourquoi ?

« Les hommes se distinguent par ce qu'ils montrent et se ressemblent par ce qu'ils cachent. »

Paul Valéry, Mélange, 1939

« Nos défauts et nos infirmités ne sont pas ridicules en eux-mêmes, mais ridicule est l'effort que nous déployons pour les dissimuler. »

Giacomo Leopardi, extrait des Pensées (1835)



À l'occasion de l'année européenne des personnes handicapées en 2003, une association belge, l'Association nationale pour le logement des handicapés, a réalisé une vidéo de sensibilisation. On y voit un groupe de personnes attablées dans un restaurant. Un homme fait la conversation, accumulant une série de clichés misogynes. On sent le malaise s'installer autour de la table, mais personne n'ose intervenir pour s'opposer à cet homme. Puis, la caméra recule sur un plan plus large et on découvre à ce moment là que l'homme qui tient ces propos insultants est en fauteuil roulant.

La vidéo se termine par une voix off qui énonce : « Cet homme est une personne handicapée. Mais avant tout, cet homme est un con. Les personnes handicapées sont des femmes et des hommes comme les autres ».

SOURCE Vidéo de l'Association nationale pour le logement des handicapés

Retrouvez cette vidéo, avec sous-titrage, sur le site Universal Subtitle à l'adresse suivante :

► <http://ur1.ca/9vo4y^a>

a. <http://universalsubtitles.org/en/videos/d5lwZs1g5Tb/>

Si le handicap renvoie à la notion de « limitation », comme le définit la loi française, alors nous pouvons dire que l'être humain est par essence handicapé. Mais bien sûr, selon le contexte, on peut l'être plus ou moins. Ainsi, une personne en fauteuil roulant et qui n'aurait plus l'usage de ses jambes n'est pas en situation de handicap quand elle surfe sur le Web. Alors qu'il lui est plus difficile d'aller faire ses courses toute seule si le magasin n'est pas accessible. De même, une personne sourde pourra faire ses courses sans problème dans ce même magasin, mais elle ne pourra pas lire de vidéo sur le Web si cette dernière n'est pas sous-titrée.

VOCABULAIRE Déficience, incapacité et handicap

La première classification de l'Organisation mondiale de la santé (OMS) en 1980 distinguait la *déficience*, qui est le dysfonctionnement d'un organe par rapport à son fonctionnement « normal », de l'*incapacité* pouvant découler de ce dysfonctionnement, et du *handicap* ou *désavantage* pouvant résulter de cette incapacité. Les trois n'étant pas nécessairement liés.

En 2001, l'OMS révisé sa classification pour mettre l'accent sur la complexité des interactions sociales et environnementales et propose quatre composantes :

- fonctions organiques ;
- structures anatomiques ;
- activités et participation ;
- facteurs environnementaux.

► <http://ur1.ca/9vo5e^a>

a. <http://apps.who.int/classifications/icfbrowser/Default.aspx>

Difficile dans ces conditions, me direz-vous, de savoir précisément qui est concerné par l'accessibilité du Web ! La question n'est pas en effet de savoir *à qui s'adresse un site accessible*, puisqu'il s'adresse potentiellement à tout le monde, mais plutôt *dans quels contextes il faut prévoir que le site soit accessible*.

Les différents types de handicaps

Combien de personnes concernées ?

Nous avons vu précédemment que la grande nouveauté accompagnant le concept d'accessibilité est la mise en contexte des situations. On devrait d'ailleurs parler, pour être exact, de personnes « en situation de handicap » plutôt que de personnes handicapées.

Cette notion de situation de handicap a été entérinée par la loi, y compris dans les dispositifs d'indemnisation et de compensation du handicap. Prenons l'exemple d'une personne devenant soudain allergique à la farine. Si cette personne est informaticienne, elle ne pourra prétendre à aucune indemnisation. En revanche, si cette même personne travaille dans une boulangerie, elle sera fondée à demander une aide et pourra être déclarée travailleur handicapé.

Le besoin de chiffrer le nombre de personnes handicapées sert avant tout aux pouvoirs publics pour mettre en place des dispositifs d'aide et définir qui y aura droit. Mais *la réalité des limitations d'activité dans la vie quotidienne touche un nombre bien plus important de personnes que la part des gens déposant un dossier de prise en charge du handicap*. D'où la difficulté à obtenir des chiffres fiables.

SOURCE INSEE

Les situations de handicap, d'incapacité ou de dépendance dépassent la seule reconnaissance administrative :

► <http://ur1.ca/9vo6d^a>

Handicap, incapacité ou dépendance reconnu administrativement, ressenti ou identifié :

► <http://ur1.ca/9vo6v^b>

a. http://insee.fr/fr/themes/document.asp?reg_id=20&ref_id=15901#trois

b. http://insee.fr/fr/themes/document.asp?reg_id=20&ref_id=15901&page=alapage/alap328/alap328_encad1.htm

Une enquête de l'INSEE (Institut national de la statistique et des études économiques), commandée par le ministère de la Santé et publiée en août 2009, s'est appuyée sur le témoignage des personnes elles-mêmes pour estimer le nombre de personnes handicapées.

SOURCE Enquête « Vie quotidienne et santé » (INSEE)

Voir le détail de l'enquête « Vie quotidienne et santé » sur le site de l'INSEE :

► <http://ur1.ca/9vo79a>

a. http://www.insee.fr/fr/themes/document.asp?ref_id=ip1254

L'INSEE conclut qu'une personne sur dix considère avoir un handicap, et qu'une personne sur cinq se déclare limitée dans ses activités.

Autrement dit, *20 % des personnes interrogées ont une limitation*, même si elles ne veulent pas pour autant parler de handicap ! C'est donc loin d'être négligeable, surtout quand son objectif est de toucher le plus d'internautes possible avec son site...

EXEMPLES Le handicap, vu d'ailleurs...

En Norvège aussi, difficile de chiffrer le nombre de personnes handicapées. Suivant les définitions, l'estimation du taux de handicap en Norvège varie de 10 % à plus de 25 % pour le même échantillon de personnes.

📖 Revue *Alter* vol. 4, n° 3 juillet-septembre 2010, Elsevier Masson SAS, p.175

La coopérative québécoise Accessibilité Web affirme par ailleurs que « les statistiques démontrent que 45,4 % de la population du Québec est susceptible d'éprouver des difficultés sur le Web lorsque les préoccupations les plus élémentaires d'accessibilité ne sont pas prises en compte. »

Voir les statistiques recensées sur le site :

► <http://ur1.ca/9vo70a>

a. <http://accessibiliteweb.com/fr/accessibilite/benefices.html>

Quels types de limitations ?

Les situations sont très variées et la façon de vivre son handicap est intimement liée à chaque personne. Étant donné cette diversité, il ne s'agira pas d'être exhaustif mais nous pouvons tenter d'identifier cinq grands types de handicaps ayant un impact sur l'utilisation d'un ordinateur.

Handicaps visuels

Si je commence par ce type de handicap, c'est parce que c'est celui auquel la plupart des gens pensent quand ils envisagent l'accessibilité du Web.

Le handicap visuel ne concerne pas que les personnes non voyantes, il regroupe également tout un ensemble de troubles visuels ou de maladies qui vont d'une baisse plus ou moins forte de la vue, à une vue tronquée ou altérée par des taches, par exemple, en passant par une perception des couleurs qui peut être déficiente.

2 – L'accessibilité : pour qui ? Pourquoi ?

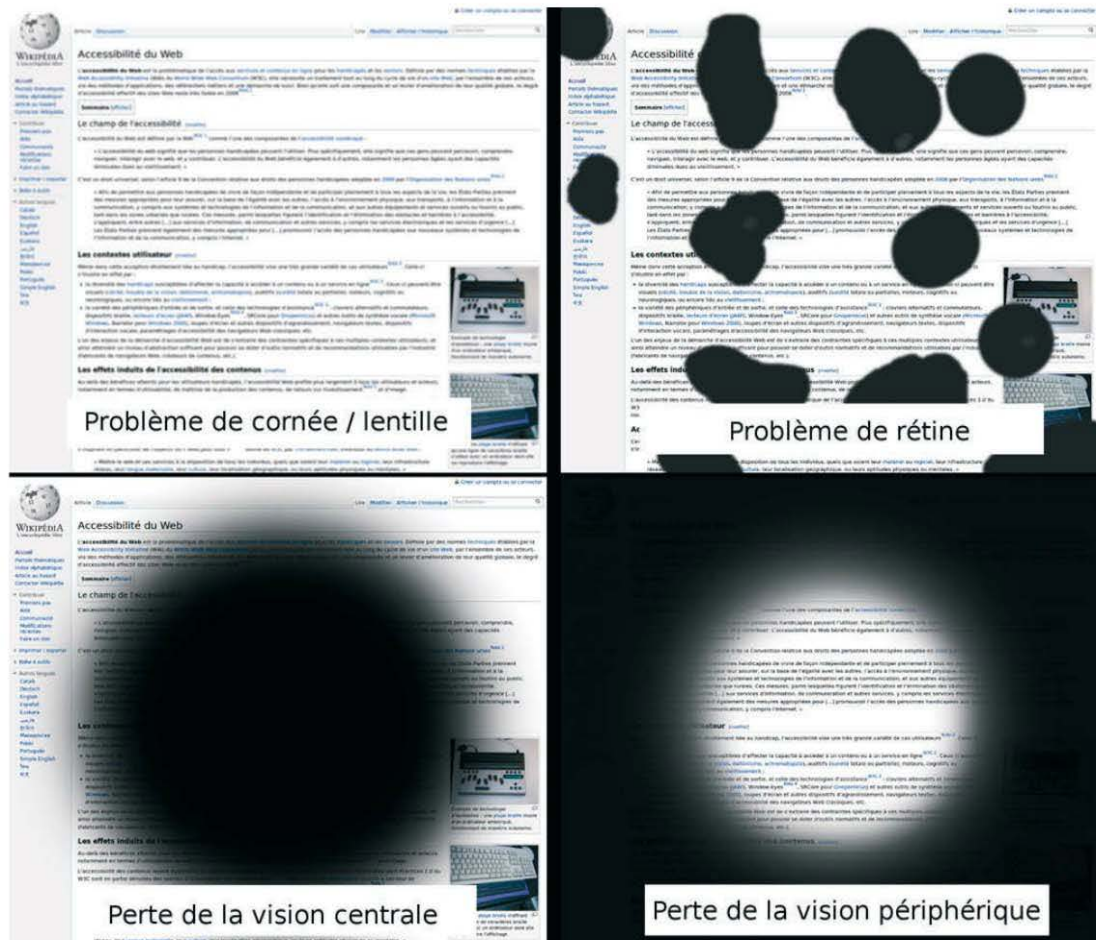


Figure 2-1 Page Wikipédia sur l'accessibilité du Web vue avec différents troubles de la vision : problèmes de cornée/lentille, problème de rétine, perte de la vision centrale et perte de la vision périphérique

RESSOURCES Simulations

Pour une simulation des différents troubles de la vision :

► <http://ur1.ca/9vo8d^d>

Pour simuler les différents types de daltonismes appliqués à une page web de votre choix :

► <http://ur1.ca/9vo8l^b>

Signalons par ailleurs la possibilité de simuler les trois types de daltonismes avec les filtres du logiciel libre de retouche d'image GIMP. Voir la procédure sur le site de documentation :

► <http://ur1.ca/9vo9c^c>

a. http://www.lowvisiononline.unimelb.edu.au/fr/LowVision/LowVision_simulation.htm#check

b. <http://www.vischeck.com/vischeck/vischeckURL.php>

c. <http://docs.gimp.org/fr/gimp-display-filter-dialog.html>

Le handicap visuel n'est pas le handicap le plus répandu, mais il associe plusieurs contraintes quand on parle informatique. Car en plus de ne pas pouvoir voir ce qui est affiché à l'écran, une personne non voyante ne pourra pas non plus utiliser la souris, puisqu'elle ne peut pas viser le pointeur sur un écran qu'elle ne voit pas.

EN RÉSUMÉ Handicap visuel

Les principaux handicaps à la navigation sur le Web pour une personne non voyante ou malvoyante sont les suivants :

- impossibilité pour certains de voir ce qui est affiché à l'écran. Il faut donc un autre moyen de restitution : oral ou tactile ;
- impossibilité d'utiliser la souris : la navigation au clavier doit être permise ;
- contraste des couleurs insuffisant ;
- certaines couleurs ne sont pas vues : il faut pouvoir modifier la feuille de styles ;
- affichage des textes dans une taille de police trop petite : il faut pouvoir agrandir les textes ;
- impossibilité de lire des textes trop gros (vision tubulaire) : il faut pouvoir réduire la taille de la police.

Handicaps auditifs

Le handicap auditif recouvre lui aussi une grande variété de situations : du sourd profond qui n'entendra aucun son à celui qui ne percevra que certains sons (graves ou aigus, par exemple), en passant par les personnes qui n'entendent qu'à un certain volume sonore ou dont l'audition est gênée par des bruits parasites...

À cela s'ajoute la difficulté que certaines personnes peuvent éprouver à l'acquisition de la lecture et l'écriture. En effet, la plupart des langues dans le monde, à l'exception du chinois, sont basées sur l'association de lettres pour former des sons. Or, quand on ne perçoit pas ces sons, le langage devient très abstrait et il est très difficile d'accéder à l'écrit, ce qui ne pose pas de problème à une personne devenant sourde à l'âge adulte.

Figure 2-2

Le site Websourd : tous les textes, y compris ceux des menus de navigation, sont associés à un équivalent en langue des signes.



C'est pour cela que la langue des signes peut s'avérer une bonne alternative aux textes traditionnels, comme sur le site Websourd (<http://websourd.org>) représenté à la figure précédente.

CULTURE Malentendant ou « malsourd » ?

Il existe une véritable « culture sourde », née avec l'invention de la langue des signes au XVIII^e siècle et réaffirmée en réaction à l'isolement dans lequel les personnes sourdes furent enfermées à la suite du Congrès de Milan de septembre 1880 où elle fut interdite. La langue des signes française (LSF) réapparaît en 1976 et est reconnue comme une langue à part entière en 2005.

Une amie sourde et professeure de langue des signes aime à plaisanter en parlant de « malsourd » plutôt que de malentendant, pour renverser la question de la normalité. À méditer...

Vous pouvez consulter l'article de Wikipédia pour en savoir plus sur le congrès de Milan :

► <http://ur1.ca/9voam^a>

a. http://fr.wikipedia.org/wiki/Congr%C3%A8s_de_Milan

EN RÉSUMÉ Handicap auditif

Les principaux handicaps à la navigation sur le Web pour une personne sourde ou malentendante sont les suivants :

- impossibilité d'accéder aux éléments sonores du site, et donc de comprendre un média vidéo ou audio si aucun texte ne l'accompagne ;
- difficulté à lire un texte pour certains, préférence pour la langue des signes.

Handicaps moteurs

Dans le cas de l'informatique, ce sont les mains qui sont nécessairement employées pour taper au clavier ou utiliser la souris. Ainsi, les personnes ayant une déficience motrice ne sont pas forcément handicapées face à un ordinateur, à moins que cette déficience concerne la mobilité des membres supérieurs. Dans certains cas, seule la navigation à la souris est rendue difficile, dans d'autres, la navigation au clavier est également impossible. Quoi qu'il en soit, l'ensemble d'un site n'est accessible que si la navigation est permise à la fois à la souris et au clavier.

EN RÉSUMÉ Handicap moteur

Les principaux handicaps à la navigation sur le Web pour une personne ayant un handicap moteur sont les suivants :

- impossibilité de naviguer à la souris, navigation au clavier nécessaire pour accéder à tout le contenu du site et activer toutes les fonctions avec le clavier ;
- difficultés avec certaines fonctions comme le glisser-déposer (*drag and drop*) qui sont réalisées à la souris.

Handicaps « mentaux »

Derrière cette appellation générique se cache une pluralité de situations n'ayant par forcément grand rapport les unes avec les autres. La définition légale distingue d'ailleurs le handicap mental des handicaps cognitif et psychique. L'objet de cet ouvrage n'étant pas de faire un panorama complet de ces handicaps, concentrons-nous sur certains d'entre eux ayant un impact direct sur l'accès au Web.

Les handicaps intellectuels

Il s'agit des personnes ayant des difficultés dans l'apprentissage et une compréhension moins rapide que la moyenne. Plus précisément, l'Union nationale des associations de parents, de personnes handicapées mentales et de leurs amis (Unapei) considère le handicap mental comme étant d'abord la conséquence sociale d'une déficience intellectuelle.

L'Unapei précise que « La personne en situation de handicap mental éprouve des difficultés plus ou moins importantes de réflexion, de conceptualisation, de communication et de décision. »

SOURCE Le handicap mental selon l'Unapei

► <http://ur1.ca/9vobj^a>

a. <http://www.unapei.org/Le-handicap-mental-sa-definition.html>

Les degrés de handicap intellectuel peuvent varier fortement, allant d'une compréhension plus lente à une incapacité totale d'accéder à la lecture et à l'écriture. Il est donc d'autant plus important pour ces internautes de les guider dans leur navigation avec des repères facilement identifiables, en évitant des événements activés sans action de l'utilisateur et qui seraient perçus comme une perturbation pouvant le conduire à quitter le site.

Les troubles DYS

Contrairement au handicap intellectuel, les troubles DYS concernent des dysfonctionnements spécifiques liés à l'apprentissage et à la connaissance, sans qu'il y ait de handicap intellectuel global.

La Fédération française des DYS définit ainsi six catégories de troubles cognitifs spécifiques :

- 1 les troubles spécifiques de l'acquisition du langage écrit, communément appelés dyslexie et dysorthographe ;

- 2 les troubles spécifiques du développement du langage oral, communément appelés dysphasie ;
- 3 les troubles spécifiques du développement moteur et/ou des fonctions visuo-spatiales, communément appelés dyspraxie ;
- 4 les troubles spécifiques du développement des processus attentionnels et/ou des fonctions exécutives, communément appelés troubles d'attention avec ou sans hyperactivité ;
- 5 les troubles spécifiques du développement des processus mnésiques ;
- 6 les troubles spécifiques des activités numériques, communément appelés dyscalculie.

SOURCE Les six catégories de troubles DYS selon la Fédération française des DYS

► <http://ur1.ca/9vod1^a>

a. <http://www.ffdys.com/troubles-dys/nature-des-troubles/introduction.htm>

Sur le Web, cela se traduit notamment par des difficultés à lire un contenu affiché à l'écran quand le texte est justifié (c'est-à-dire aligné) des deux côtés, la difficulté à se concentrer sur un texte quand des animations sont activées, ou encore par la difficulté à résoudre un captcha logique.

VOCABULAIRE Captcha

Un captcha est un système anti-spam. Vous voyez ? Ces codes qu'on vous demande d'entrer soit en recopiant des lettres déformées soit en réalisant un calcul simple ou en répondant à une question pour prévenir les spams, etc.

Pour en savoir plus sur les captchas et l'accessibilité, voir page 99.

L'épilepsie photosensible

Il existe plusieurs types d'épilepsies. Celle qui concerne particulièrement la navigation sur le Web est l'épilepsie « photosensible ». L'association Épilepsie France en donne la définition suivante :

« L'épilepsie photosensible généralisée est une forme particulière d'épilepsie provoquée par la stimulation lumineuse (télévision, vidéo ou jeux électroniques, reflet du soleil sur la neige ou l'eau, stroboscope des discothèques, voire les motifs réguliers d'un papier peint, d'un vêtement, d'un radiateur de chauffage central ou d'un escalier mécanique). Dans la majorité des cas, il existe exclusivement des crises provoquées par la lumière. »

SOURCE Définition de l'épilepsie photosensible par l'association Épilepsie France

► [http://ur1.ca/9voe5^a](http://ur1.ca/9voe5a)

a. <http://www.epilepsie-france.com/index.php/comprendre/les-syndromes-epileptiques/epilepsie-photosensible-generalisee.html>

Ainsi, les animations clignotantes avec des couleurs criardes, outre le mal de tête qu'elles peuvent occasionner à n'importe qui, sont également susceptibles de provoquer des crises d'épilepsie !

J'entends déjà vos interrogations et objections...

– Oui, bon d'accord, c'est peut-être possible en théorie, mais qui aujourd'hui conçoit encore des sites comme ça ? Et puis, ce doit être vraiment très rare, a-t-on au moins réellement recensé des cas de crises d'épilepsie provoquées par un site web (non, mais, franchement ?!) ?

Eh bien oui, et récemment en plus ! Un exemple ? Le site des Jeux Olympiques de Londres 2012. Il s'agissait en fait d'une vidéo de promotion des JO de Londres avec une animation à l'image du logo jaune fluo et fuchsia (!). Décriée pour des questions esthétiques dès sa sortie, cette animation aurait provoqué plusieurs crises d'épilepsie. Les organisateurs ont donc décidé de retirer l'animation.

EXEMPLE Le site des JO de Londres 2012, cause de crises d'épilepsie

Actualité sur le site de la BBC :

► [http://ur1.ca/9voeo^a](http://ur1.ca/9voeo3a)

Lien vers le site officiel actuel des JO de Londres 2012 :

► [http://ur1.ca/9vof3^b](http://ur1.ca/9vof3b)

a. <http://news.bbc.co.uk/2/hi/6724245.stm>

b. <http://www.london2012.com/fr/>

EN RÉSUMÉ Handicap « mental »

Les principaux handicaps à la navigation sur le Web pour une personne ayant l'un des handicaps « mentaux » mentionnés précédemment sont les suivants :

- difficultés de compréhension d'un texte écrit ;
- difficultés à effectuer des calculs ;
- impossibilité de supporter des animations ou flashes trop rapides et lumineux (crises d'épilepsie) ;
- difficultés de concentration et d'attention ;
- réalisation de certaines tâches plus lentement que la moyenne (besoin de maîtriser son environnement, de lancer et arrêter les animations et vidéos, etc.) ;
- grande fatigue à la lecture sur écran.

Troubles de santé invalidants

Bien qu'il ne s'agisse pas d'une catégorie de handicap différente de celles mentionnées précédemment, je tenais à vous rappeler la grande variété des situations en pointant également les troubles de santé invalidants.

Certaines maladies nécessitent des traitements lourds aux effets secondaires très handicapants : troubles de la concentration, grande fatigue, fourmillements et parfois même perte de l'usage de certains membres (notamment lors du traitement de certaines formes du cancer). L'état peut être seulement temporaire, il n'en est pas moins invalidant.

De façon moins dramatique, il suffit de se casser le bras pour être incapable de se servir de sa souris normalement. Prendre en compte l'accessibilité dans la conception d'un site web, c'est autant d'opportunités de toucher un public aussi large que varié. Et si vous pensez à vos internautes, il y a des chances pour qu'ils vous le rendent en restant plus longtemps sur vos pages (à condition bien sûr que le contenu soit intéressant, mais c'est une autre histoire...).

À LIRE Les handicaps invisibles

Article sur les handicaps invisibles à découvrir sur le site du Fil Santé Jeunes et extrait du livre *Les handicaps invisibles, comment les identifier, les combattre et les surmonter* de H. Rubinstein, paru aux éditions du Seuil en 2008. On y lit notamment que « la plupart des incapacités reprises sous le terme de handicap invisible sont en fait des comportements "habituels" devenus handicapants par leur intensité et/ou leur fréquence d'apparition ».

► <http://ur1.ca/9vog6^a>

Par ailleurs, dans un article du *Figaro* du 15 novembre 2010 intitulé « Le chantier du "handicap invisible" pour les entreprises », la journaliste Aude Seres affirme que 80 % des handicaps en entreprises sont invisibles.

► <http://ur1.ca/9vohi^b>

a. <http://www.filsantejeunes.com/dossiers-classes-par-categories/dossier-dossier-du-mois/6521>

b. <http://www.lefigaro.fr/emploi/2010/11/15/01010-20101115ARTFIG00780-le-chantier-du-handicap-invisible-pour-les-entreprises.php>

Handicap et vieillissement

L'évolution démographique des pays développés va vers un vieillissement des populations. Ainsi, l'INSEE prévoit qu'en France, en 2050, un habitant sur trois sera âgé de 60 ans ou plus, contre un sur cinq en 2005.

Si le vieillissement n'est pas forcément synonyme de handicap, il n'est pas rare de voir une diminution de certaines capacités quand on prend de l'âge. Baisse de la vue, de l'audition, arthrose, etc., sont autant de symptômes nous indiquant que le temps a passé.

SOURCE Projections de population pour la France métropolitaine à l'horizon 2050

► <http://ur1.ca/9voi0a>

a. http://www.insee.fr/fr/themes/document.asp?ref_id=ip1089

Le Web étant devenu un outil incontournable dans le quotidien de nombreuses personnes, ce n'est pas parce qu'elles vieillissent qu'elles ne l'utiliseront plus. Prendre en compte l'accessibilité dans la conception de sites web bénéficie donc également aux personnes âgées. Ce qui fait un potentiel, selon les projections de l'INSEE, de 23 millions de personnes de plus de 60 ans en France, en plus des personnes handicapées de moins de 60 ans. Si l'on vous prétend que l'accessibilité du Web ne concerne qu'une petite partie des internautes, vous saurez donc quoi répondre !

Maintenant que nous venons de voir quelles étaient les difficultés principales que des personnes handicapées pouvaient éprouver sur le Web, voyons comment la technologie peut aider à surmonter certains handicaps.

Les technologies d'assistance

Pour comprendre comment faire de l'accessibilité web, il faut commencer par comprendre ce qu'est une technologie d'assistance. Dans un sens très large, c'est une technologie qui vient s'ajouter à la configuration de base de l'ordinateur pour faciliter l'usage de l'une de ses « fonctions ».

Reprenons notre analogie avec le bâtiment. Pour monter à l'étage d'un immeuble, vous pouvez emprunter les escaliers. C'est la fonction de base nécessaire pour monter. Mais certains immeubles sont équipés d'un ascenseur. L'ascenseur peut être considéré comme une technologie d'assistance. Une personne en fauteuil roulant ne pourra pas accéder à l'étage sans ascenseur, mais tout le monde trouvera bien pratique de l'utiliser, surtout si on a une valise à transporter, par exemple. C'est le principe de l'accessibilité, valable pour le Web : elle est nécessaire pour certains, mais profitable à tous.

EXEMPLE Le handicap, vecteur d'innovation ?

« L'innovation systématique requiert la volonté de considérer le changement comme une opportunité », disait Peter Drucker, auteur de référence sur le management d'entreprise. Imposant un dépassement de soi, le handicap est justement source d'innovation.

Saviez-vous que la télécommande de télévision avait été inventée à l'origine pour les personnes à mobilité réduite ? Qui saurait s'en passer aujourd'hui ?

De même, l'invention des lunettes de soleil est attribuée à James Ayscough, en 1752. En voulant corriger des troubles de la vision avec une lumière bleue ou verte, il créa les premières lunettes aux verres teintés.

Autre exemple, dans le domaine de l’informatique : la souris. Pour fonctionner, un ordinateur n’a besoin que d’une unité centrale et d’un clavier. Or, la souris est bien une technologie qui s’ajoute à cette configuration de base pour faciliter l’utilisation de l’ordinateur.

EXEMPLE L’écran, technologie d’assistance pour non-aveugle ?

En poussant le raisonnement, on peut conclure que l’écran, non indispensable dans le fonctionnement d’un ordinateur, est une technologie d’assistance pour les personnes pouvant le voir. En effet, une personne non voyante n’utilise pas d’écran, puisqu’elle ne peut pas le voir. Elle fera usage d’un autre moyen de restitution : un logiciel de lecture d’écran couplé à une synthèse vocale et/ou une plage Braille.

Ce qu’il est important de comprendre, c’est que *l’accessibilité du Web ne signifie pas développer des « options » d’accessibilité pour telle ou telle catégorie de personnes*. Elle consiste à permettre aux internautes de naviguer sur le Web de la façon qui leur convient. *Un site web accessible est un site web compatible avec les technologies d’assistance*. Il n’appartient pas au développeur web de créer des gadgets pour pallier des déficiences : l’utilisateur qui en a besoin est équipé.

VOCABULAIRE Technologie d’assistance ou aide technique ?

Si depuis la traduction en français des règles d’accessibilité WCAG 2, le terme « technologie d’assistance » tend à s’imposer, vous entendrez encore souvent parler « d’aide technique ». Il s’agit en réalité de la même chose. Vous pourrez également rencontrer l’abréviation « A.T. », en référence à l’anglais *assistive technology*.

Pour une définition plus spécifique des technologies d’assistance dans le cadre des règles d’accessibilité des contenus web (WCAG 2), nous pouvons reprendre celle proposée par le W3C : « matériel ou logiciel qui agit comme agent utilisateur ou simultanément avec un agent utilisateur usuel afin de fournir des fonctionnalités répondant aux besoins des utilisateurs ayant des limitations fonctionnelles, fonctionnalités qui vont au-delà de celles qui sont offertes par les agents utilisateurs usuels ».

SOURCE Définition de « technologie d’assistance » par le W3C

► <http://ur1.ca/9voib>^a

a. <http://www.w3.org/Translations/WCAG20-fr/#atdef>

Là, évidemment, vous vous demanderez : « Mais c’est quoi, au juste, un agent utilisateur ? » Ce à quoi le W3C répond : « Tout logiciel qui récupère et présente le contenu web aux utilisateurs. Exemple : les navigateurs web, les lecteurs de médias,

les modules d'extensions et les autres programmes – dont les technologies d'assistance – qui aident à récupérer, restituer et interagir avec le contenu web. »

SOURCE Définition de « agent utilisateur » par le W3C

► <http://ur1.ca/9voir^a>

a. <http://www.w3.org/Translations/WCAG20-fr/#useragentdef>

Bref, si vous ne goûtez pas le style alambiqué du W3C, on peut résumer en disant qu'une technologie d'assistance est une technologie (matériel ou logiciel) qui permet de pallier certaines déficiences. Nous n'allons pas dresser la liste de tout ce qui existe, mais en énumérer quelques-unes pour vous donner une idée concrète de ce dont il s'agit.

Logiciels de lecture d'écran et synthèse vocale

Pour une personne aveugle ou malvoyante au point d'être dans l'incapacité de voir l'écran, il est non seulement nécessaire de pouvoir avoir connaissance du contenu des pages web par un autre moyen, mais également de pouvoir naviguer, se repérer et effectuer des actions.

C'est là qu'interviennent les logiciels de lecture d'écran (appelés aussi revues d'écran). Ces logiciels doivent pallier le handicap visuel et remplacer la vue en restituant les textes présentés à l'écran, mais également leur nature. Ils analysent donc la nature des éléments HTML pour la restituer.

Le logiciel agit comme le cerveau en analysant des données. Mais il doit être associé à un moyen de restituer ces données, soit par une synthèse vocale (qui sera la voix du logiciel), soit par une plage Braille, que nous verrons ci-après. Couplé à une synthèse vocale, l'ordinateur va donc « parler » à l'internaute pour lui communiquer des éléments essentiels tels que la structure de la page (titre de niveau 1, titre de niveau 2, liste à puces...), les actions possibles (lien, case à cocher non cochée, bouton Valider...), et même des informations invisibles aux utilisateurs voyants, récupérées dans le code de la page (une alternative à une image, par exemple).

Le logiciel de lecture d'écran le plus connu est Jaws, édité par la société Freedom Scientific¹. C'est en effet le logiciel le plus employé, fonctionnant sous environnement Windows, car souvent financé par les maisons départementales des personnes handicapées (MDPH). Une version de démonstration vous permet de l'utiliser pendant 40 minutes, après quoi il vous faut redémarrer. Mais la licence de cette version n'autorise pas une utilisation dans un contexte de tests.

1. <http://www.freedomsci.de/serv01fra.htm>

EXEMPLE L’ordinateur des aveugles

Pour une démonstration de l’utilisation d’un logiciel de lecture d’écran couplé à une synthèse vocale, vous pouvez visionner la vidéo produite par l’université de Nice-Sofia Antipolis :

► <http://ur1.ca/9vojc^a>

Vous pouvez voir également une autre démonstration (en anglais) en vidéo sur le site du logiciel libre de lecture d’écran NVDA :

► <http://ur1.ca/9voj^b>

a. <http://www.youtube.com/watch?v=Kz1AQ4dqxgA>

b. <http://www.nvda-project.org>

À SAVOIR Extrait des Conditions générales d’utilisation de Jaws (version 13.0.977)

« LICENCE DE DÉMONSTRATION ET D’ÉVALUATION SEULEMENT POUR UN USAGE NON COMMERCIAL

Les licences de logiciels qui nécessitent un redémarrage du système au bout d’un certain temps ou qui expirent au bout d’un certain nombre de jours sont censées être utilisées à titre d’évaluation avant acquisition uniquement. Tout usage commercial est interdit. Vous n’êtes, en aucune circonstance, en tant qu’utilisateur final, autorisé, en capacité d’exploiter dans un but commercial ce LOGICIEL, ou une portion de celui-ci. Ni vous, ou qui que ce soit sous vos ordres, ni quelque tiers que ce soit, ne devez entreprendre les actions suivantes à l’égard de ce LOGICIEL, ou d’une portion de celui-ci, à savoir, vendre, prêter, louer, ni exploiter commercialement ni utiliser ce LOGICIEL en vue d’un objectif commercial. Par exemple, *ces licences d’évaluation et de démonstration ne sont pas destinées au développement ni aux tests de scripts JAWS, d’applications, de codage HTML, ni d’autres codes Web.* »

Il existe des alternatives libres et performantes qui vous permettront de tester l’utilisation d’une revue d’écran gratuitement et en toute légalité. Parmi elles, citons NVDA, logiciel libre pour Windows, ou Orca, sous environnement Gnome pour le système d’exploitation GNU/Linux.

TÉLÉCHARGEMENT NVDA et Orca

Pour télécharger la version française de NVDA :

► <http://ur1.ca/9vokd^a>

Site officiel (en anglais) pour télécharger Orca :

► <http://ur1.ca/9vokn^b>

a. <http://www.nvda-fr.org/download.php>

b. <http://live.gnome.org/Orca/DownloadInstall>

Vous êtes maintenant équipé pour faire parler votre ordinateur ? Bien, encore faut-il maintenant que les sites web fonctionnent de façon à ce que le logiciel puisse récupérer les informations nécessaires... Nous verrons comment faire dans la deuxième partie du livre.

GADGET Les options « lire le texte »

Comme nous venons de le voir, les logiciels de lecture d'écran ne se contentent pas de lire le texte à l'écran, mais restituent également une foule d'informations sur le contenu et la structuration de la page web qui permet à l'utilisateur de naviguer librement (de lien en lien, de titre en titre, etc.). Toutes ces informations sont données par le logiciel dont sont équipés les internautes qui en ont besoin.

Ainsi, les lecteurs audio présents sur certains sites et proposant de lire le texte d'un article ne peuvent remplacer ces technologies d'assistance. Il s'agit plutôt d'un « gadget » qui a sans doute un intérêt (marketing, ergonomique...) mais ce genre d'options ne doit surtout pas être considéré comme étant de l'accessibilité. Souvenez-vous : l'accessibilité du web n'est pas optionnelle, elle doit être disponible pour tous. Un site web accessible n'a pas besoin de proposer des « options » réservées à certains types de handicaps.

Logiciels de lecture d'écran et plage Braille

Nous venons d'évoquer l'emploi de logiciels de lecture d'écran couplés à une synthèse vocale. Or, il est également possible de coupler le logiciel à une plage Braille pour une restitution tactile plutôt que vocale. Voici quelques exemples de ces plages Braille en images.



Figure 2-3 Différents modèles de plages Braille : les picots se lèvent et s'abaissent pour composer des lignes de texte en Braille au fur et à mesure de la lecture.

(Crédits Matthieu Faure, photos en CC-BY-SA sur Wikimedia Commons : <http://ur1.ca/a8buo>^a)

a. <http://commons.wikimedia.org/wiki/User:Mfaure>

Logiciels de grossissement

Pour les personnes pouvant voir leur écran mais ayant des difficultés à lire des textes trop petits, il existe des logiciels de grossissement ou « loupe », qui n’ont rien à voir avec les pictogrammes représentant une loupe qu’on trouve parfois sur certains sites.

GADGET Les pictos loupe ou A⁺/A⁻

De même que pour les lecteurs audio permettant de lire le texte d’un article, les options représentant une loupe ou un A majuscule suivi des signes + et – pour agrandir et réduire la taille du texte sont également des gadgets dont l’intérêt n’est pas l’accessibilité mais, au mieux, la valorisation de celle-ci.

Ce n’est pas parce qu’un site propose cette option qu’il est accessible et, inversement, un site n’a pas besoin de proposer cette option pour être accessible. Non pas qu’il ne faille pas proposer ce genre d’outils, mais il faut simplement avoir conscience qu’en faisant cela, on ne s’occupe pas d’accessibilité.

L’illustration suivante (figure 2-4, page suivante) présente une comparaison entre un agrandissement × 6 via l’option « loupe » proposée sur le site du Comité interministériel du handicap, l’agrandissement maximal de ce même site avec le raccourci clavier **Ctrl+** de Firefox et un agrandissement avec la loupe Compiz, sous Ubuntu. Avec l’option « loupe », on voit que seul le texte central est grossi, mais pas les menus, tandis que les différentes boîtes ne se redimensionnent pas : la page est illisible. Au contraire, les logiciels de grossissement comme Orca permettent un agrandissement de tout l’espace visible sur l’ordinateur et pas seulement de la page web.

SOURCE Site du Comité interministériel du handicap

Lien vers la page testée :

► <http://ur1.ca/9vol5^a>

a. <http://www.social-sante.gouv.fr/espaces,770/handicap,775/institutionnel,803/le-comite-interministeriel-du,1711/le-comite-interministeriel-du,11306.html>

Afin de tester ces logiciels, vous pouvez télécharger, par exemple, le logiciel libre Virtual Magnifying Glass (VMG) ou utiliser Orca, cité précédemment. De nombreux paramètres sont disponibles dans ce genre de logiciels.



Figure 2-4 Comparaison entre différents modes d'agrandissement sur le site du Comité interministériel du handicap : en haut à gauche, la page sans agrandissement ; en haut à droite, un agrandissement maximal avec le bouton A+ disponible sur le site (le résultat est illisible et ne permet d'agrandir que le contenu, pas les menus) ; en bas à gauche, un agrandissement à 200 % avec Firefox (permet de tout agrandir, mais une mauvaise conception du site fait se chevaucher les blocs) ; enfin, un agrandissement avec la loupe de Compiz sur Ubuntu, seul moyen de vraiment agrandir toute la page.
(<http://ur1.ca/abz20a>)

a. <http://www.social-sante.gouv.fr/actualite-presse,42/communiqués,95/espaces,770/handicap,775/espaces,770/handicap,775/institutionnel,803/le-comite-interministeriel-du,1711/le-comite-interministeriel-du,11306.html>

TÉLÉCHARGEMENT Logiciel d'agrandissement et autres technologies d'assistance libres

Pour télécharger VMG :

► <http://ur1.ca/9volj>^a

Vous retrouverez aussi un ensemble de technologies d'assistance libres sur la page wiki du groupe de travail de l'April « Accessibilité et logiciels libres ». Ce projet est en cours et ouvert à tous, n'hésitez pas à y contribuer !

► <http://ur1.ca/9volp>^b

a. <http://magnifier.sourceforge.net>

b. http://wiki.april.org/w/Technologies_d%27assistance_libres

Headstick et pointeurs laser

Pour pallier l'incapacité de certains internautes à se servir de leurs bras, des solutions matérielles existent tels que les *headsticks* ou *headpointers*, extension cintrée sur la tête permettant de taper sur un clavier avec un « bâton » (figure 2-5). On trouve également des pointeurs laser, là encore fixés sur la tête et permettant un contrôle de son ordinateur par la tête, en le reliant à un logiciel de saisie de texte.

Figure 2-5

Utilisateur se servant d'une licorne (headstick ou headpointer, en anglais) pour taper sur son clavier (Crédits : Karl Dean, créateur de <http://www.ablehere.com>)



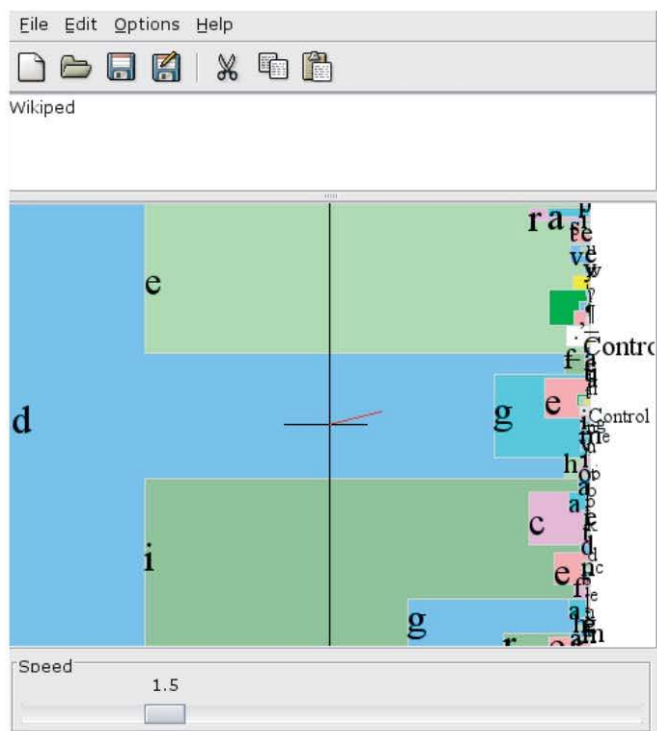
Logiciels de saisie de texte et claviers virtuels

Les logiciels de saisie de texte et les claviers virtuels peuvent avoir plusieurs fonctions : permettre à une personne avec un handicap moteur de taper du texte, en le couplant, par exemple, avec un pointeur laser ; faciliter la saisie de textes par des personnes handicapées mentales ou avec un trouble DYS ; proposer des alternatives aux claviers traditionnels (en LSF, avec des pictogrammes, etc.).

Voici un exemple de ce type de logiciel : avec Dasher, logiciel libre de saisie de texte, le texte se réorganise de façon prédictive pour faciliter la saisie et la rendre plus rapide pour les personnes qui l'utilisent, par exemple, avec un pointeur laser.

Figure 2-6

Le logiciel Dasher en action



Autre exemple : le logiciel libre Chewing Word propose plusieurs versions de son clavier afin de faciliter la communication selon différents profils.

Figure 2-7

Le clavier prédictif de Chewing Word, le mâcheur de mots



TÉLÉCHARGEMENT Dasher et Chewing Word

Deasher est un logiciel libre à télécharger sur le site du projet :

▶ <http://ur1.ca/9vomc>^a

Également logiciel libre, Chewing Word peut être téléchargé sur le site dédié :

▶ <http://ur1.ca/9vomi>^b

a. <http://www.inference.phy.cam.ac.uk/dasher/>

b. <http://chewingword.wikidot.com>

Et le handicap auditif alors ?

Concernant le handicap auditif, hormis les claviers virtuels en langue des signes, les technologies d'assistance sont surtout des logiciels permettant de moduler les sons, rien de spécifique à la navigation web. Ou bien il s'agit d'un recours à des interfaces de médiation via une webcam. Mais ces « interfaces de médiation » sont des personnes, pas des technologies, qui interprètent en temps réel ce que la personne dit afin de lui permettre, par exemple, d'appeler le médecin ou d'effectuer des démarches par téléphone.

Ainsi, le handicap auditif est évidemment un handicap à prendre en compte, mais davantage dans la conception des contenus et dans le fait de mettre à disposition des versions texte des contenus sonores que pour l'utilisation d'une technologie spécifique.

À LIRE Guide : l'informatique au service des handicapés

Ce livre de Patrick Collignon, paru en 2003 aux éditions Eyrolles, est relativement ancien et ne concerne pas précisément l'accessibilité du Web, mais il détaille les différents dispositifs permettant aux personnes handicapées d'utiliser un ordinateur – ce qui n'est pas l'objet de notre livre.

 Patrick Collignon, *L'informatique au service des handicapés : guide de référence*, Paris, OEM-Eyrolles, 2003, 296 pages.

Handicaps, technologies d'assistance... L'accessibilité web se concentre sur tout un ensemble de besoins très spécifiques. Or, avant d'être handicapée, n'oublions pas que la personne qui va visiter votre site est d'abord un internaute...

Des besoins spécifiques, comme tout le monde !

Comme nous l'avons vu précédemment, les concepts même de handicap et d'accessibilité ne sont pas neutres. Ils ont été influencés par l'histoire et continuent d'évoluer. Il en va de même concernant l'accessibilité du Web : ce qui est considéré aujourd'hui comme relevant de l'accessibilité du Web ne l'était pas forcément hier, et certains critères évoluent également dans l'autre sens, rendant possibles des choses qui, hier encore, étaient considérées comme inaccessibles.

Pour mieux comprendre, nous pouvons faire un parallèle avec la différence qui existe entre soins de santé (remboursés par la Sécurité Sociale) et soins de confort. Ce qui détermine le remboursement est largement conditionné par l'évolution sociale, la situation économique, mais aussi par des considérations technologiques (un médicament générique moins cher et aboutissant au même résultat). Ainsi, un même médicament peut-il être considéré comme indispensable et être remboursable à un certain moment, puis ne plus l'être pour insuffisance du service médical rendu à un autre moment. De

même, entre les versions 1 et 2 des règles d'accessibilité du Web (WCAG 1 et 2), dix ans se sont écoulés et plusieurs évolutions ont vu le jour. Nous verrons cela en détail dans le chapitre 4, « Comprendre les règles d'accessibilité du Web ».

FICTION Un fauteuil roulant volant

Faisons un peu de science-fiction. Dans un avenir pas si lointain, un inventeur met au point une nouvelle technologie permettant aux fauteuils roulants de s'élever dans les airs au contact d'une bande magnétique. Tous les escaliers équipés de cette bande magnétique pourraient être empruntés par ces fauteuils nouvelle génération. Ainsi, les escaliers deviendraient accessibles aux fauteuils roulants, à condition que les utilisateurs soient équipés de ces fauteuils dernier cri et que les propriétaires d'escaliers se soient dotés de ces bandes magnétiques.

Pour le Web et les technologies d'assistance, c'est la même chose. Avec les évolutions technologiques, ce qui était inaccessible hier peut devenir accessible aujourd'hui. Reste à déterminer le point critique qui permettra de dire que le nombre d'utilisateurs équipés de la technologie dernière génération et le nombre de lieux ou espaces web accessibles est suffisant pour ignorer ceux qui ne sont pas équipés.

Cette évolution donne régulièrement lieu à des débats entre spécialistes de l'accessibilité et ergonomes pour savoir ce qui entre dans leur champ de compétences respectif. Nous n'aurons pas la prétention de trancher un débat en évolution permanente, mais nous pouvons dire que, dans tous les cas, accessibilité et ergonomie sont complémentaires.

L'accessibilité est l'élément de base à prendre en compte pour permettre un accès à tous les internautes, et nous avons vu que la variété des situations de handicap rend cette question pertinente quel que soit le profil utilisateur ciblé. L'ergonomie web consiste, quant à elle, à rendre un site web utile et facile à utiliser pour les visiteurs de ce site.

Ainsi, si l'accessibilité est une nécessité de base, elle est très insuffisante pour faire un site de qualité. Plus qu'un gage de succès de votre site, il faut la voir comme l'élément premier essentiel qui, s'il n'est pas pris en compte, vous fera perdre des visiteurs à coup sûr.

COMPRENDRE Accessibilité et ergonomie

Vous pouvez lire l'article en anglais « Accessibility is not enough » de Jakob Nielsen, éminent spécialiste de l'utilisabilité, sur son site :

► <http://ur1.ca/9vomx^a>

a. <http://www.useit.com/alertbox/accessibility.html>

La tentation du test utilisateur en guise d'évaluation de l'accessibilité

Si accessibilité et ergonomie sont complémentaires et que la limite entre accessibilité et utilisabilité est parfois difficile à déterminer, il s'agit cependant de deux domaines très différents.

Penser que faire un test utilisateur avec une personne handicapée permettra d'évaluer l'accessibilité d'un site web est une erreur. Car comme nous l'avons vu, il y a autant de façons d'utiliser et de paramétrer sa technologie d'assistance que de personnes.

S'il est vrai que de plus en plus d'ergonomes prennent en compte l'accessibilité et que, inversement, l'accessibilité peut fonder une démarche d'ergonomie, il est important de ne pas confondre ces deux domaines de compétence.

Pour un même handicap, l'internaute non expert pourra trouver le site plus ou moins accessible. Sans compter qu'étant donné la variété des handicaps, il est risqué de concevoir un site en se basant sur le retour exclusif de personnes atteintes d'un handicap particulier.

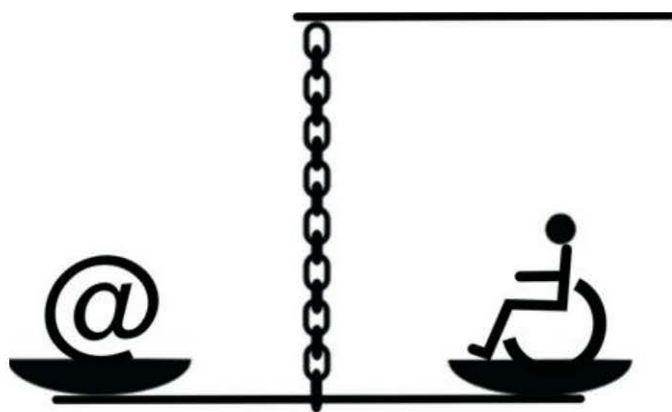
C'est pour cela qu'ont été élaborées des règles internationales pour l'accessibilité des contenus web, les WCAG. Elles se basent sur la pratique des utilisateurs et reposent donc sur une réalité concrète. Rédigées de façon à n'oublier personne, en croisant les études et les analyses, elles dépassent donc le cadre du test utilisateur effectué pour un site web donné. Ce sont ces règles d'accessibilité qui font figure de référence dans la loi française.

Contexte technologique et juridique

*« Selon que vous serez puissant ou misérable,
Les jugements de cour vous rendront blanc ou noir. »*

Jean de La Fontaine¹

Pour poursuivre notre analogie avec le bâtiment, il est intéressant de savoir, en plus des enjeux humains, sur quelles normes il est possible de s'appuyer pour prendre en compte l'accessibilité de nos sites web.



1. *Les Fables*, Livre VII *Les animaux malades de la peste*.

Évolutions du Web et des technologies d'assistance

Vers un Web « sémantique » ?

HTML5, CSS3... Depuis la création du groupe de travail WAI en 1997, le Web n'a cessé de changer, avec une accélération notable des transformations ces dernières années, devenant un terrain de jeux et d'apprentissage constant pour les passionnés !

Les évolutions ne sont pas que technologiques ; elles concernent aussi les pratiques des concepteurs de sites qui tendent à se professionnaliser. Abandonnant petit à petit les mises en page exclusivement constituées de tableaux HTML et le mélange entre la forme et le fond, les développeurs se soucient davantage de la qualité de leur code.

VOCABULAIRE La forme et le fond

Notons que cette expression, bien que passée dans le langage courant, exprime mal ce qu'elle représente. Par « fond », on entend les contenus web (textes, images, éléments multimédias ou autres) mais également leur structuration.

La « forme » concerne la présentation des éléments de contenu.

C'est dans ce sens que nous utiliserons les termes de fond et de forme dans l'ensemble de l'ouvrage.

On doit cela évidemment au travail du W3C, qui a une forte représentation à Sophia-Antipolis notamment, mais aussi à des communautés comme Alsacrérations, qui ont largement contribué à populariser l'usage des feuilles de styles, par exemple, au profit de l'accessibilité.

SOURCE Alsacrérations

Alsacrérations est une communauté d'apprentissage pour les standards du Web. Son fondateur, Raphaël Goetter, est l'auteur de livres sur l'utilisation des feuilles de styles CSS. Vous pouvez consulter en particulier le forum de discussion dédié aux questions d'accessibilité :

► <http://ur1.ca/9vonk^a>

a. <http://forum.alsacreations.com/list-6-Accessibilite-du-Web.html>

En effet, comme nous avons pu l'entrevoir dans le chapitre précédent, les technologies d'assistance interprètent essentiellement le code HTML, c'est-à-dire la nature des éléments de contenu dans une page web, et non sa mise en page. Il est donc important de bien séparer le fond et la forme. C'est pourquoi, afin de pouvoir restituer, par exemple, la structure d'une page, encore faut-il qu'elle ait été structurée sémantiquement (c'est-à-dire selon son sens et non selon l'apparence qu'on veut lui donner), à l'aide de balises de titres, notamment.

VOCABULAIRE Web sémantique

Pour reprendre la définition introductive donnée par Wikipédia : « Le Web sémantique désigne un ensemble de technologies visant à rendre le contenu des ressources du World Wide Web accessible et utilisable par les programmes et agents logiciels, grâce à un système de métadonnées formelles, utilisant notamment la famille de langages développés par le W3C ».

► <http://ur1.ca/9vonw^a>

Ainsi, le fait d'utiliser les balises pour structurer une liste non ordonnée (`ul` et `li`) plutôt que de faire des tirets à la main permettra aux technologies d'assistance d'indiquer à l'utilisateur qu'il s'agit d'une liste composée de n items, par exemple.

Pour une vision futuriste de ce que peut devenir le Web sémantique, lisez l'article de Tim Berners-Lee, James Hendler et Ora Lassile, traduit en français par Élisabeth Lacombe et Jo Link-Pezet :

► <http://ur1.ca/9vooe^b>

a. http://fr.wikipedia.org/wiki/Web_s%C3%A9mantique

b. <http://www.urfist.cict.fr/archive/lettres/lettre28/lettre28-22.html>

Cette évolution est donc positive, même s'il reste beaucoup de travail pour que ces bonnes pratiques se généralisent. Mais dans le même temps, les technologies web évoluent, plus rapidement que les technologies d'assistance, et posent également de nouveaux défis pour l'accessibilité.

Les défis du Web riche pour l'accessibilité

Le Web évolue de plus en plus vers un Web applicatif, permettant aux internautes de participer et d'accéder à des fonctionnalités toujours plus sophistiquées. Il est loin le temps où « site web » était synonyme de « site statique ». Aujourd'hui, un site web moderne est très souvent un site web dynamique, auquel il est très facile de contribuer sans avoir à saisir du code, un site web qui vit, qui évolue, qui interagit avec les réseaux sociaux, qui permet aux utilisateurs de participer...

Ces mises à jour en temps réel sont notamment rendues possibles grâce à Ajax (*Asynchronous JavaScript and XML*), qui associe deux technologies, le JavaScript et le XML, pouvant s'actualiser de façon « asynchrone », c'est-à-dire non simultanée. Ainsi, vous pouvez consulter un site web (en HTML ou XML) et avoir des informations qui vont se mettre à jour automatiquement (via JavaScript) sans que vous ayez besoin de recharger toute la page.

Or, imaginez une personne aveugle utilisant une revue d'écran. Comment saura-t-elle que de nouvelles informations sont apparues sur la page ? À l'heure où nous écrivons ces lignes, les technologies d'assistance interprètent le HTML. Si la page est modifiée via JavaScript et n'est pas rechargée, les technologies d'assistance sauront qu'il y a eu un changement, mais ne pourront pas identifier sa nature ni ses répercussions sur le reste de la page. C'est donc à l'auteur, via un moyen quelconque et plus

particulièrement via ARIA (voir ci-après), de transmettre toutes les informations nécessaires sur la nature et le comportement des contenus insérés dynamiquement.

IDÉE REÇUE « Je ne veux pas un site accessible, je veux un site beau et dynamique ! »

S'il est vrai que les évolutions du Web, avec le développement d'applications web riches, posent des défis pour l'accessibilité, cela ne signifie pas que l'on doive renoncer à utiliser ces techniques pour faire accessible !

Affirmer que l'accessibilité empêche de réaliser des sites modernes, dynamiques et esthétiques est un non-sens absolu !

Aujourd'hui, il existe de nombreuses solutions permettant de proposer des fonctionnalités aussi esthétiques et dynamiques qu'accessibles. La bonne pratique consiste avant toute chose à rechercher ces solutions, car du choix de ces outils dépend aussi l'accessibilité du résultat final.

Avant de vous creuser la tête, pensez à vérifier qu'il n'existe pas déjà des scripts prévoyant l'accessibilité. Vous serez étonnés de voir la créativité des développeurs partageant leur travail !

Et si aucune solution n'est possible, une petite alternative fera très bien l'affaire. Ce n'est peut-être pas toujours simple, mais c'est toujours possible.

Pour s'en convaincre, il suffit de regarder la galerie des sites web ayant reçu le label AccessiWeb :

▶ <http://ur1.ca/9vop0^a>

a. <http://www.accessiweb.org/index.php/galerie.html>

Par ailleurs, avec la popularisation de bibliothèques JavaScript telles que jQuery ou Prototype, l'apparition de nouveaux effets pas toujours accessibles au clavier nativement envahissent la Toile. Carrousel, glisser-déposer, etc., peuvent poser des difficultés.

C'est pour répondre à ces nouvelles manières de faire du Web qu'un groupe de travail s'est penché sur l'élaboration d'une nouvelle spécification, WAI-ARIA, visant à ajouter des attributs aux balises HTML pour fournir aux auteurs (développeurs) un moyen simple de dialoguer avec les technologies d'assistance en leur fournissant toutes les informations nécessaires sur des éléments inconnus dans le langage HTML. Trois types d'informations peuvent être fournis : le rôle (ceci est un *slider*), l'état (ce sous-menu est ouvert) et des propriétés (ce sous-menu contient *x* éléments). De cette manière, le développeur peut expliquer de manière très simple comment tel ou tel contenu doit être restitué et quel doit être son comportement. Nous verrons cela en détail dans le chapitre 12, « HTML5, ARIA et accessibilité ».

Cette spécification prometteuse n'étant pas encore suffisamment implémentée par les technologies d'assistance, une alternative à JavaScript est toujours demandée selon les méthodes d'évaluation françaises permettant de vérifier la conformité aux

règles d'accessibilité internationales. Mais WAI-ARIA apporte une réponse astucieuse à une question très actuelle, et plusieurs navigateurs et technologies d'assistance ont déjà commencé à l'implémenter.

Accessibilité web et obligation légale

C'est aux confins du droit et de la technique que nous plonge l'accessibilité du Web. L'exercice qui consiste, par exemple, à savoir si le critère 7.5 (au hasard) du référentiel AccessiWeb 2.2 est applicable dans tel contexte, est tout à fait comparable à certains cas pratiques que les étudiants en droit doivent résoudre ! Quant à l'application concrète de l'accessibilité, elle dépend beaucoup de décisions politiques et légales nécessaires à une prise de conscience. Commençons par étudier les aspects légaux.

Le développement de l'accessibilité par la loi

Dans un monde parfait, les personnes en charge de la commande ou de la conception de sites web, en particulier quand celles-ci travaillent dans la fonction publique et ont pour mission de garantir l'intérêt général, ne pourraient imaginer créer un site sans qu'il soit accessible à tous.

Or, nous ne vivons pas dans un monde parfait... C'est donc sous la pression des associations de personnes handicapées que les législateurs de différents pays ont mis en place un dispositif législatif, plus ou moins tardivement, plus ou moins élaboré, plus ou moins contraignant.

Dans la foulée d'une communication sur l'e-accessibilité faite par la Commission européenne en 2005, une étude pour « mesurer les progrès de l'accessibilité numérique en Europe » fut publiée en 2007, mise à jour en 2011.

Il en ressort trois grandes conclusions :

- 1 Le niveau d'accessibilité numérique global en Europe est très mauvais.
- 2 Les pays européens, dans leur ensemble, sont loin derrière des pays comme l'Australie, le Canada ou les États-Unis.
- 3 Il existe de très grandes disparités dans la façon de prendre en compte l'accessibilité numérique parmi les États membres de l'Union européenne.

Notons que l'obligation légale d'accessibilité en France n'est pas accompagnée et donc peu opérante. Ce que l'étude MeAC confirme en classant la France de manière très médiocre en termes de résultats.

SOURCE Étude MeAC – Measuring progress of eAccessibility in Europe

Pour télécharger le rapport MeAC (en anglais), consultez le site suivant :

► <http://ur1.ca/a405d^a>

a. <http://www.eaccessibility-monitoring.eu/researchResult.aspx>

Cependant, malgré ce constat peu brillant, un point intéressant est à relever : le lien qui existe entre la mise en place d'une politique en faveur de l'accessibilité numérique et la prise en compte effective de l'accessibilité dans la conception de sites web. Car, comme nous le disions en introduction, l'accessibilité du Web tient d'abord et avant tout au fait de l'avoir prévue en amont dans les critères de conception du site web. Bref, encore faut-il y penser et pour cela, encore faut-il savoir que ça existe ! En ce sens, les dispositifs législatifs, même s'ils ne sont pas toujours contraignants, semblent bénéfiques en termes de sensibilisation. Voyons ce qu'il en est pour la France.

Le contexte législatif français

La loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées

En France, c'est l'article 47 de la loi n° 2005-102 du 11 février 2005 qui pose le principe d'une obligation d'accessibilité pour les sites web publics :

« Les services de communication publique en ligne des services de l'État, des collectivités territoriales et des établissements publics qui en dépendent doivent être accessibles aux personnes handicapées.

L'accessibilité des services de communication publique en ligne concerne l'accès à tout type d'information sous forme numérique quels que soient le moyen d'accès, les contenus et le mode de consultation. Les recommandations internationales pour l'accessibilité de l'Internet doivent être appliquées pour les services de communication en ligne. »

SOURCE Article 47 de la loi du 11 février 2005

Consultez la page suivante du site Légifrance pour prendre connaissance de cet article dans son intégralité :

► <http://ur1.ca/9voqc^a>

a. <http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000000809647&dateTexte=#LEGIARTI000006682279>

Nous pouvons retenir trois éléments.

- Seul le secteur public est concerné par cet article de loi, ce qui touche tout de même un grand nombre de structures, ainsi que les agences web privées réalisant le site web de ces administrations.
- L'obligation ne concerne pas que le Web, mais toute ressource à disposition sur un site web.
- Ce sont les « recommandations internationales pour l'accessibilité de l'Internet » qui font référence. Autrement dit, les règles publiées par le groupe de travail WAI du W3C.

Ces principes étant énoncés, plusieurs questions se posent : comment sait-on si la loi est respectée, autrement dit, si les sites web publics sont accessibles ? Comment s'effectue le contrôle ? Qui est chargé de faire respecter la loi ? Que se passe-t-il si la loi n'est pas respectée et que le site n'est pas accessible ? Quel niveau d'accessibilité faut-il atteindre ?

À NOTER L'obligation légale d'accessibilité, également dans la LCEN

La LCEN (Loi pour la confiance dans l'économie numérique) est antérieure à la loi Handicap puisqu'elle date du 21 juin 2004. Or, elle pose dans son article 3 le principe suivant : « L'État, les collectivités territoriales, les établissements publics et les personnes privées chargées d'une mission de service public veillent à ce que l'accès et l'usage des nouvelles technologies de l'information permettent à leurs agents et personnels handicapés d'exercer leurs missions. ». Cela dépasse même le cadre du Web.

Le texte de la loi n° 2004-575 du 21 juin 2004 pour la confiance dans l'économie numérique est disponible sur le site Légifrance à l'adresse suivante :

► <http://ur1.ca/a41qq^a>

a. http://www.legifrance.gouv.fr/affichTexteArticle.do;jsessionid=6B20C1A7CB53C2A8551FD2AD287A45E8.tpdjo05v_2?idArticle=LEGIARTI000006421543&cidTexte=LEGITEXT000005789847&dateTexte=20120827

Sans réponse à ces questions, la loi est inapplicable. C'est donc avec une grande impatience que nous attendions le décret d'application devant rendre cet article de loi effectif. Et l'attente dura... quatre ans !

AILLEURS L'accessibilité web, une obligation légale dans d'autres pays

USA : section 508

► <http://ur1.ca/9voqj^a>

Québec : standards du gouvernement du Québec sur les ressources informationnelles SGQRI 008

► <http://ur1.ca/9voqt^b>

a. <http://www.section508.gov>

b. <http://www.msg.gouv.qc.ca/normalisation/index.html>

Le décret d'application

Sans volonté politique, le moindre petit texte peut prendre une éternité à être rédigé et publié. Je vous laisse estimer celle-ci à l'aune des quatre années nécessaires pour obtenir ce décret tant attendu. Et finalement, en ce beau jour du 14 mai 2009, le décret n° 2009-546 apporta enfin quelques réponses... tout en posant de nouvelles questions.

SOURCE Décret n° 2009-546 du 14 mai 2009

Le texte du décret n° 2009-546 est disponible sur le site Légifrance à l'adresse suivante :

► <http://ur1.ca/9vorf³>

a. <http://www.legifrance.gouv.fr/affichTexte.do?dateTexte=&categorieLien=id&cidTexte=JORFTEXT000020616980>

Composé de six articles, voilà ce que nous pouvons retenir de ce décret :

- Pour savoir comment évaluer l'accessibilité des sites web, il faudra se reporter à un référentiel. Référentiel qui sera déterminé... par arrêté.
- Un délai pour se mettre en conformité avec la loi est fixé à trois ans à partir de la publication du décret pour les collectivités et services s'y rattachant, délai ramené à deux ans pour les services de l'État. Autrement dit, ce délai a déjà expiré pour tout le monde.
- Pour savoir si une démarche de mise en accessibilité a été entreprise, il est demandé aux éditeurs de sites publics de publier une attestation de conformité. Or, il est précisé que cette attestation sera délivrée par « une autorité administrative compétente », sans préciser de quelle autorité il s'agit.
- Le ministre en charge et les préfets ont la responsabilité de constater le défaut de conformité. La sanction prévue pour les sites contrevenant à la loi consiste à figurer sur une liste noire.
- Enfin, l'article 6 pose le principe d'une *obligation de formation pour les agents*. Cette formation doit comporter un enseignement théorique et pratique sur l'accessibilité des services de communication publique en ligne aux personnes handicapées et sur la conformité aux règles et standards nationaux et internationaux en cette matière.

L'arrêté ministériel relatif au référentiel général d'accessibilité pour les administrations : que faut-il faire concrètement ?

Pour que le dispositif législatif soit complet et que la loi soit enfin applicable, il a donc fallu attendre la publication de l'arrêté ministériel relatif au Référentiel général d'accessibilité pour les administrations (RGAA), le 21 octobre 2009.

SOURCE Arrêté ministériel du 21 octobre 2009

Consultez l'intégralité du texte de cet arrêté sur le site Légifrance, référence NOR : BCFJ0917114A :

► <http://ur1.ca/9vorr^a>

a. <http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000021208630&fastPos=1&fastReqlId=2012857444&categorieLien=cid&oldAction=rechTexte>

Il fixe le RGAA dans sa version 2.2.1 comme étant la méthode d'évaluation de référence pour vérifier la conformité avec les règles internationales.

Il désigne également le délégué interministériel aux personnes handicapées comme étant en charge de l'exécution de cet arrêté. À noter que ce dernier n'existe plus, la délégation ayant été remplacée par le Comité interministériel des personnes handicapées.

Le Référentiel général d'accessibilité pour les administrations introduit en réalité un ensemble de documents, tous disponibles sur le site de la Direction générale de modernisation de l'État (DGME), dont une partie concerne les aspects législatifs, et l'autre, l'application technique des règles d'accessibilité.

SOURCE Site du RGAA

N'hésitez pas à consulter le site du Référentiel général d'accessibilité pour les administrations à l'adresse suivante :

► <http://ur1.ca/653em^a>

a. <http://www.references.modernisation.gouv.fr/rgaa-accessibilite>

Nous parlerons donc plutôt de « dispositif RGAA », comprenant :

- le document RGAA servant d'introduction générale à l'accessibilité du Web et présentant les documents du dispositif ;
- le guide d'accompagnement, remplaçant le RGAA dans son contexte législatif et détaillant les modalités d'application ;
- les annexes 1, 2 et 3 qui constituent la partie technique et qui présentent respectivement les critères de succès, les tests de conformité et une grille de correspondance entre critères de succès et tests de conformité. Nous verrons cela en détail dans la deuxième partie.

Dans la mesure où cette première partie de l'ouvrage traite déjà de ce qu'est l'accessibilité du Web, nous ne nous attarderons pas sur le document RGAA à proprement parler. Et comme nous gardons les documents techniques pour la deuxième partie, penchons-nous plutôt sur le guide d'accompagnement.

Le guide d'accompagnement du RGAA explique de façon non technique les obligations à respecter. La section 5, page 18 de ce guide est intitulée « Modalités d'appli-

cation du RGAA ». On y expose notamment le niveau de conformité à respecter, la déclaration de conformité à publier et les dérogations possibles.

Niveau de conformité à respecter

Les règles internationales d'accessibilité du Web, définies par la WAI, s'appellent WCAG 2. Sur le modèle de WCAG 2, le référentiel d'accessibilité est organisé en trois niveaux : A (niveau basique), AA (niveau normal) et AAA (niveau d'amélioration). Ces niveaux sont cumulatifs, c'est-à-dire que pour être conforme au niveau AA, il faut valider les critères de niveau A et ceux de niveau AA.

Le guide d'accompagnement indique :

« Le niveau recommandé par l'Union Européenne est le niveau AA. *C'est également le niveau attendu pour les sites concernés par le RGAA et à ce titre, pour être conforme au RGAA, il est nécessaire de valider l'ensemble des tests ayant un niveau WCAG déduit A et AA.* Les critères de succès associés au niveau AAA peuvent être pris en compte dans certains contextes, lorsque cela est possible et pertinent. »

Dans un monde parfait, depuis mai 2012, tous les sites publics devraient donc être accessibles en respectant le niveau AA du RGAA. Mais une fois encore, nous ne sommes pas dans un monde parfait, l'accessibilité des sites publics est tellement rare dans les faits qu'il est impossible de ne pas constater un échec dramatique en la matière.

Comment cela se fait-il ? Bien que les raisons soient sans doute multiples et que nous n'ayons pas fait d'études approfondies pour répondre sérieusement à cette question, nous pouvons évoquer quelques pistes.

- 1 L'absence d'information et de sensibilisation sérieuse.** Encore trop souvent, dans de nombreuses collectivités et administrations, les agents ignorent totalement ce qu'est l'accessibilité numérique.
- 2 Le manque de formation aussi bien des acheteurs que des concepteurs et développeurs web.** Quand, dans un cahier des charges public, une mention indique discrètement que la refonte ou la création du site devra respecter le RGAA, rares sont ceux qui le respectent effectivement. Les acheteurs publics ne savent que trop rarement évaluer le sérieux des réponses, et une fois le travail accompli, personne ne vérifie que le niveau d'accessibilité est réellement atteint. Quant à ceux qui réalisent le site, beaucoup manquent encore cruellement de formation et pensent parfois bien faire, grâce à diverses lectures, mais sans en avoir pour autant compris toutes les nuances.
- 3 Un manque de volonté politique ?** Cette remarque est peut-être un peu provocatrice, mais elle est malheureusement incontestable. Tous les budgets publics souffrent de la crise, mais même sans cela, à budget limité, il faut faire des choix. Or l'accessibilité est rarement prioritaire. Et même si son coût est à relativiser

puisqu'il s'agit avant tout d'un coût de formation et éventuellement d'accompagnement, et que ce coût peut être également considéré comme un investissement et à mettre en balance avec les bénéfices que cela apporte, rien n'y fait. Vous pensez que ce portrait dressé est un peu trop sombre ? Comptez le nombre de sites accessibles, ou ayant tout simplement publié une déclaration de conformité et entamé une démarche de mise en accessibilité... La loi date de 2005, et depuis la publication du décret, trois ans se sont écoulés. Un délai qui permettait de faire beaucoup mieux que ce que nous avons aujourd'hui.

Par cet ouvrage, nous espérons contribuer modestement à combler les lacunes mentionnées aux points 1 et 2, mais pour ce qui est de la volonté politique, la balle est dans le camp des décideurs... mais également dans celui des personnes handicapées et des associations qui décideront ou non de faire respecter leurs droits.

L'attestation de conformité

Le guide d'accompagnement indique que les administrations, collectivités et établissements publics qui en dépendent devront publier une « attestation de conformité au RGAA » détaillant les résultats des tests effectués pour vérifier l'application du RGAA, justifier les manquements aux règles et expliquer leur politique d'accessibilité afin de combler les lacunes dans l'avenir.

EN PRATIQUE Évaluer la conformité au RGAA

L'attestation de conformité doit indiquer le niveau de conformité au RGAA. Mais rien n'empêche l'évaluateur d'employer une autre méthode d'évaluation de conformité au RGAA. Ainsi, il est tout à fait possible de procéder à l'évaluation avec le référentiel AccessiWeb 2.2, par exemple, ou tout autre référentiel, pour être conforme au RGAA. Nous verrons cela plus en détail dans la deuxième partie du livre.

Cette déclaration de conformité au RGAA est censée être publiée dès la mise en ligne du site et consiste en un « engagement sur l'honneur de satisfaire à l'ensemble des tests (sauf dérogation dûment justifiée) ayant un niveau WCAG déduit A et AA dans la version du RGAA en vigueur ». Cette attestation est donc une autodéclaration venant expliquer la démarche d'accessibilité entreprise et détaillant le niveau d'accessibilité atteint à un instant t .

Les objectifs de cette attestation, décrits à la page 23 du guide d'accompagnement, sont les suivants :

- démontrer les avancées de leur démarche de mise en conformité,
- signaler les futures améliorations du site,
- montrer aux utilisateurs que les problèmes sont connus et anticipés,

- faire remonter au ministère chargé des personnes handicapées les problèmes de mise en œuvre sur le terrain,
- faciliter l'ajustement ou la mise en œuvre des actions de formation, de sensibilisation ou de mise à disposition d'outils. »

Bref, au-delà d'une simple attestation, il s'agit d'un véritable plan de route pour l'accessibilité du Web.

En pratique, il n'existe pas de modèle d'attestation de conformité. Mais certaines mentions sont obligatoires, notamment :

- la date de réalisation ;
- la version du RGAA de référence ;
- le nom et l'adresse e-mail du déclarant ;
- les technologies employées sur le site ;
- la liste des agents utilisateurs et technologies d'assistance utilisées pour vérifier l'accessibilité des contenus ;
- la liste des pages du site ayant fait l'objet de la vérification de conformité ;
- le résultat des tests et la justification des dérogations.

Une évaluation de l'accessibilité du site devra être réalisée avant la mise en ligne, en vue d'établir l'attestation de conformité. Cette évaluation doit porter sur un échantillon de pages représentatif du site, qui varie en général entre une dizaine et une vingtaine de pages.

Parmi les pages qui doivent faire l'objet d'une évaluation, doivent figurer impérativement :

- la page d'accueil du site, la page contact et les mentions légales ;
- la page accessibilité, reprenant les informations de l'attestation de conformité, la politique d'accessibilité retenue et les évolutions prévues sur le site en matière d'accessibilité ;
- la page d'aide, qui contient des informations pour aider les internautes à comprendre le fonctionnement du site, indiquant les principaux services et contenus fournis ; cette page d'aide peut également être fusionnée avec la page accessibilité ;
- le plan du site et la recherche ;
- les pages d'accès aux contenus ou fonctionnalités principaux (par exemple, les rubriques de premier niveau dans l'arborescence) ;
- des pages représentatives des types de contenus disponibles sur le site (par exemple, une page contenant des tableaux de données, des éléments multimédias, des illustrations, des formulaires, etc.) ;
- les pages ayant le plus grand nombre de visiteurs.

Une fois cet échantillon représentatif sélectionné, il conviendra de procéder à l'évaluation et de donner les résultats en indiquant :

- les contenus pour lesquels le service s'engage dès maintenant sur leur accessibilité ;
- les contenus pour lesquels le service s'engage sur un calendrier de travail ;
- les raisons pour lesquelles certains tests ou certains contenus ne sont pas accessibles, avec une justification par l'administrateur du site dans son attestation.

À SAVOIR Utilisation non commerciale du RGAA

Notons une aberration juridique. Alors que le RGAA est un document officiel imposé par décret, sa licence n'est pas libre d'utilisation ! Elle est en effet restreinte à un *usage non commercial*, comme tous les documents figurant sur le site de la DGME. En effet, les mentions légales du site indiquent à la section « Droits de reproduction et liens », que :

« La reproduction de contenus figurant sur ce site est autorisée pour un usage non commercial à condition d'en indiquer la source.

De même, la mise en place de liens vers ce site, y compris "profonds", est autorisée. Nous vous prions seulement de nous informer par courriel d'une éventuelle reproduction ou pose de liens en écrivant au webmestre (page Contact). »

► <http://ur1.ca/9vosm^a>

a. <http://references.modernisation.gouv.fr/mentions-legales>

Déroptions et exceptions de conformité au RGAA

Il existe trois cas pour lesquels une dérogation au principe d'accessibilité est possible.

- 1 Un volume trop important**, rendant la mise en accessibilité totale impossible techniquement dans le délai imparti au RGAA. Cette dérogation n'est possible que dans une durée limitée et définie dans l'attestation de conformité. Ainsi, il n'est pas possible de s'exonérer indéfiniment de son obligation de conformité parce que son site présente trop de contenus. Cette dérogation permet seulement de reporter le délai de façon à ce que la mise en accessibilité soit techniquement possible.
- 2 Une obsolescence des contenus**. Cette dérogation concerne particulièrement les archives légales, que toute administration ou collectivité doit conserver dans le cadre de leur mission de sauvegarde à titre patrimonial de l'Internet. C'est le cas par exemple des scans des comptes-rendus de conseils municipaux. Ce périmètre est donc très restreint.
- 3 Un contenu tierce-partie**. C'est une dérogation étroitement liée au fonctionnement du Web, qui se veut toujours plus participatif et privilégie les interactions entre différents contenus issus de différentes sources. Ainsi, si l'auteur d'un commentaire sur un article ne respecte pas certaines règles d'accessibilité, l'administrateur du site ne peut pas en être tenu responsable. De même, lorsque le site fait appel à des contenus externes, pour la dématérialisation de certaines démarches comme la demande d'acte

d'état civil. C'est un cas courant, le service proposé étant un service souvent lui-même issu d'un acteur public. Or, si ce service n'est pas accessible, cela peut faire l'objet d'une dérogation figurant dans l'attestation de conformité.

Dans tous ces cas, le guide d'accompagnement du RGAA indique que :

- « - Les contenus non accessibles doivent être signalés à l'utilisateur pour qu'il soit informé de la proportion des pages concernées et de leur localisation dans les rubriques du site.
- Les dérogations en matière d'accessibilité doivent être expliquées et motivées dans le cadre de l'attestation de conformité.
- Il est nécessaire de prévoir un canal permettant aux personnes handicapées ou qui détectent un problème dans ce domaine de pouvoir le signaler aux administrateurs du service en ligne. »

Le guide d'accompagnement rappelle à juste titre qu'en dépit de ces dérogations, il faudra garantir l'accès à l'information recherchée par la personne handicapée, par un autre moyen s'il le faut.

L'article 47 de la loi n° 2005-102 est un petit article dans un cadre légal plus large pour la participation des personnes handicapées. L'article 11 de la loi de 2005 indique en effet que « la personne handicapée a droit à la compensation de son handicap quels que soient l'origine et la nature de sa déficience, son âge ou son mode de vie. »

À SAVOIR **L'argument financier rejeté par le Conseil d'État**

Le 22 juin 2012, le Conseil d'État a rendu une décision importante, faisant jurisprudence. Si elle ne concerne pas directement l'accessibilité numérique prévue par l'article 47, elle apporte un éclairage utile sur l'obligation d'accessibilité prévue par la loi de 2005 et l'on peut penser que sa jurisprudence est applicable également dans le cadre de l'accessibilité du Web.

Alors qu'une collectivité avait adopté dans le schéma directeur d'accessibilité une dérogation à la mise en accessibilité de certains arrêts des transports publics, en motivant cette dérogation par un coût disproportionné par rapport au service rendu (les arrêts en question étant peu fréquentés), le Conseil d'État a rejeté cette dérogation. Il a en effet estimé que « outre une centaine d'autres points d'arrêts pour lesquels une impossibilité technique existerait » sur les 1 030 que la collectivité prévoyait d'écarter de l'obligation d'accessibilité, le « motif que l'aménagement de l'ensemble du réseau aurait un coût global trop élevé » n'est pas fondé. Ainsi, seule une motivation technique peut être retenue, l'argument financier n'étant pas recevable à moins d'être réellement hors du budget de la collectivité.

Si l'on considère que cette jurisprudence est transposable concernant l'article 47, il risque de ne plus y avoir aucune excuse pour ne pas rendre un site accessible. À bon entendeur...

► <http://ur1.ca/9vosu>^a

a. <http://www.legifrance.gouv.fr/affichJuriAdmin.do?oldAction=rechJuriAdmin&idTexte=CETATEXT000026052824&fastReqId=1408881052&fastPos=1>

Quelle obligation d'accessibilité en dehors du secteur public ?

Qu'en est-il pour le secteur privé ? Les administrations et les collectivités sont-elles les seules concernées ?

À SAVOIR L'accessibilité, au-delà des frontières : investir pour être prêt

Il y a des obligations réelles et contraignantes dans un nombre croissant de pays de par le monde, notamment en Amérique du Nord et en Europe. Si l'État français est en retard, les acteurs du Web, notamment les entreprises, ont au moins trois bonnes raisons de prendre sérieusement en considération l'accessibilité :

En Europe, le plan e2020, avec par exemple :

► <http://ur1.ca/a406n^a>

Aux États-Unis, le *Twenty-First Century Communications and Video Accessibility Act* :

► <http://ur1.ca/a406z^b>

Aux Nations Unies, la convention CRPD (*Convention on the Rights of People with Disabilities*) :

► <http://ur1.ca/a407f^c>

Tôt ou tard, la France devra se mettre au niveau des autres pays. Le meilleur parti à prendre est donc de s'y préparer, ce que fait déjà le marché international.

a. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:0636:FIN:FR:HTML>

b. <http://www.fcc.gov/encyclopedia/twenty-first-century-communications-and-video-accessibility-act-0>

c. <http://www.un.org/disabilities/default.asp?id=150>

Si l'on se réfère uniquement à la loi de 2005 et à l'article 47 en particulier, il semblerait que ce soit le cas. Mais en dehors de cette loi, il existait déjà en droit français des dispositions pour définir et condamner la discrimination. Or, quand un site n'est pensé que pour certains publics, en excluant d'autres, peut-on considérer cela comme une discrimination ? Si aucune jurisprudence n'existe en France, faute de plainte et de jugement, des tribunaux étrangers ont répondu par l'affirmative en se fondant également sur la qualification de discrimination.

Ainsi, l'article 225-2 du Code pénal dispose que :

« La discrimination définie à l'article 225-1, commise à l'égard d'une personne physique ou morale, est punie de trois ans d'emprisonnement et de 45 000 euros d'amende lorsqu'elle consiste :

1° À refuser la fourniture d'un bien ou d'un service ; [...]

Lorsque le refus discriminatoire prévu au 1° est commis dans un lieu accueillant du public ou aux fins d'en interdire l'accès, les peines sont portées à cinq ans d'emprisonnement et à 75 000 euros d'amende. »

On est loin de la sanction symbolique de la publication d'une liste noire prévue par le RGAA, les sanctions, prison et amendes, étant beaucoup plus lourdes. Et dans ce

Accessibilité web

cadre là, toute personne, physique ou morale, relevant du secteur public ou privé, est concernée...

Ceci étant dit, comment se prémunir contre une plainte éventuelle ? En améliorant l'accessibilité de ses pages web bien sûr ! Et nous allons voir comment le faire, concrètement, dans la partie suivante.

DEUXIÈME PARTIE

Aspects techniques de l'accessibilité du Web

Dans la partie précédente, nous avons présenté en quoi consiste l'accessibilité. Passons maintenant à la pratique ! Vous concevez et développez peut-être vous-même des sites web ? Ou bien vous travaillez sur le Web et êtes soucieux de savoir comment dialoguer concrètement sur la question de l'accessibilité avec ceux qui administrent des sites ? Cette partie est faite pour vous !

L'accessibilité partout, pour tous, sans exception, peut paraître utopique. Il est à ce titre difficile de savoir par où commencer, compte tenu de toutes les applications dans lesquelles l'accessibilité est une question cruciale. Pour guider les concepteurs de sites, le W3C a élaboré les règles d'accessibilité pour les contenus web – les WCAG (chapitre 4), auxquelles nous allons nous référer. Cet outil précieux a donné naissance à des déclinaisons locales pour en faciliter la compréhension et l'application. Parmi elles, on trouve le référentiel AccessiWeb (chapitres 5 à 11). Nous verrons également, au-delà d'AccessiWeb, ce qu'il en est de l'accessibilité à l'heure de HTML5 (chapitre 12). Enfin, et parce que les contenus web contiennent des liens vers des ressources au-delà du Web, nous évoquerons rapidement le fait que l'accessibilité ne s'arrête pas à WCAG.

Comprendre les règles d'accessibilité du Web

« L'homme raisonnable s'adapte au monde, l'homme déraisonnable s'obstine à essayer d'adapter le monde à lui-même. Tout progrès dépend donc de l'homme déraisonnable. »

George Bernard Shaw¹

WCAG
WAI-ARIA W3C
RGAA
ACCESSIWEB

1. Extrait de *Maximes pour révolutionnaires* (1905).

Dans la jungle des normes et référentiels, on peut vite se perdre. Il est néanmoins important de connaître ce panorama normatif afin de pouvoir utiliser et d'évaluer les règles d'accessibilité, manuellement ou à l'aide d'outils lorsque cela est possible.

À SAVOIR Qui fait référence en accessibilité ?

WCAG, AccessiWeb, RGAA... cette profusion de termes ne doit pas faire oublier que dans de nombreux pays, dont la France, la référence en matière d'accessibilité émane du groupe de travail WAI du W3C. Il s'agit des règles regroupées sous l'abréviation WCAG (*Web Content Accessibility Guidelines*).

Un seul standard, les WCAG

Les WCAG (*Web Content Accessibility Guidelines*), que l'on prononce « wécague », représentent les règles d'accessibilité pour les contenus web. Elles sont élaborées collectivement par le groupe de travail WAI du W3C dont nous avons parlé dans le chapitre précédent.

Comme toute règle, les WCAG sont appelées à évoluer si cela s'avère nécessaire. Elles ne sont pas reconnues partout, certains États ayant développé leur propre corpus de règles. Cependant, elles font référence dans un grand nombre de pays et ont le mérite d'avoir été élaborées collectivement, y compris avec les utilisateurs ou en s'inspirant de recommandations issues de travaux de recherche sur les utilisateurs, et de bénéficier d'un retour d'expériences important, ce qui leur confère une certaine maturité.

Loin d'avoir le statut de dogme intangible, les WCAG doivent être abordées comme un outil dont on peut se servir pour améliorer la prise en compte de l'accessibilité de ses contenus web.

Si vous décidez de vous écarter de certaines des règles WCAG, ne le faites pas à la légère ; assurez-vous d'y avoir bien réfléchi. Car privilégier une règle plutôt qu'une autre, sur la base d'une expérience utilisateur, par exemple, peut induire des conflits d'accessibilité pour d'autres types de handicaps. Or c'est une des grandes forces des WCAG que d'avoir pensé à tous les types de handicaps.

Et si votre souci est de respecter la loi, sachez qu'en France, l'article 47 de la loi du 11 février 2005 consacre les « recommandations internationales pour l'accessibilité de l'Internet », c'est-à-dire les WCAG.

La création du groupe de travail WAI par le W3C

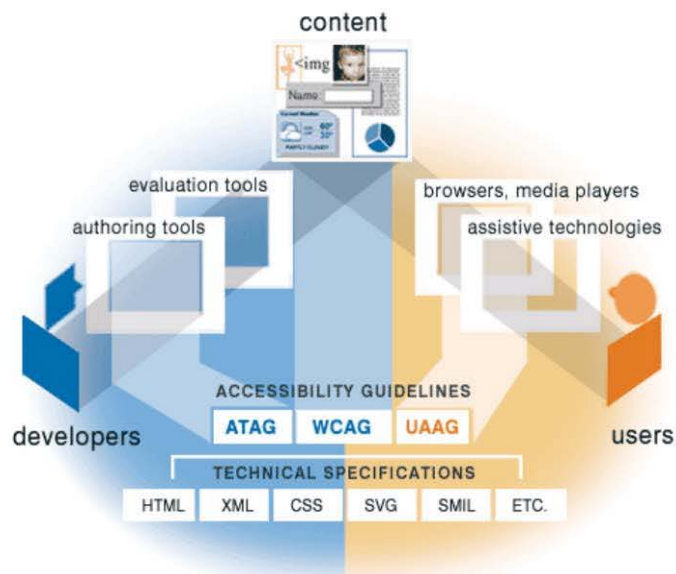
Reprenons le fil de notre histoire concernant la naissance de l'accessibilité du Web. Nous sommes en 1997, le groupe de travail WAI est créé. Dans la droite ligne des initiatives du W3C, c'est par l'élaboration de règles que la WAI va promouvoir l'accessibilité du Web.

Le Web implique plusieurs composantes qui interagissent. Les contenus web sont créés par des développeurs mais aussi par des contributeurs via un logiciel de gestion de contenus. Ils sont placés sous forme de code sur le serveur, lequel est envoyé à l'internaute à sa demande, puis interprété par un navigateur qui restitue le contenu à l'écran. Et c'est ce contenu qui est ensuite interprété par les technologies d'assistance.

Parmi les différents acteurs en présence, nous avons donc les développeurs de sites, mais aussi les contributeurs, les développeurs de navigateurs, les développeurs de logiciels de gestion de sites web (CMS) et les fabricants et développeurs de technologies d'assistance. Il suffit que l'un des maillons soit rompu pour que l'accessibilité puisse être remise en question.

Figure 4-1

Illustration des différents acteurs œuvrant pour la création et la restitution des contenus web, inspirée du graphique publié sur le site du groupe de travail WAI (<http://ur1.ca/a8c3a>)



a. <http://www.w3.org/WAI/intro/components.php>

Dans ces conditions, on comprend la volonté de la WAI, et du W3C en général, de promouvoir des standards du Web libres et ouverts afin de faciliter l'interopérabilité entre toutes ces composantes.

Dans cet ouvrage, nous nous plaçons plutôt du point de vue de l'utilisateur final, c'est-à-dire de l'internaute qui va se rendre sur votre site et des besoins qu'il va avoir pour

accéder à vos contenus. C'est donc des WCAG que nous allons parler. Mais il existe d'autres règles destinées aux autres acteurs du Web (ATAG pour les solutions d'édition ou UAAG pour les développeurs de navigateurs et de technologies d'assistance).

L'objectif des WCAG est de guider les concepteurs et développeurs de sites web en leur fournissant un moyen de vérifier si leur travail est accessible et, dans la majorité des cas, de leur proposer des solutions dont la compatibilité avec les technologies d'assistance a été vérifiée. La version actuelle des WCAG est la version 2, publiée le 11 décembre 2008.

INFO Validation des WCAG et normalisation ISO

Comme pour toute recommandation ou spécification du W3C, les WCAG suivent un processus de validation très précis, dont les différentes étapes sont les suivantes :

1. Document de travail (*Working Draft*).
2. Dernière relecture du document de travail (*Last Call Working Draft*).
3. Recommandation candidate (*Candidate Recommendation*).
4. Recommandation proposée (*Proposed Recommendation*).
5. Recommandation.

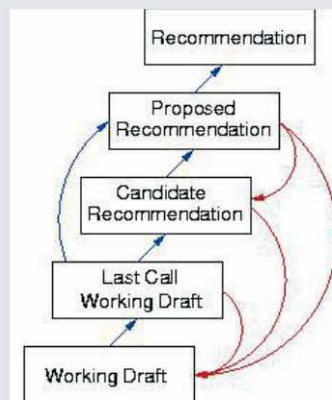


Figure 4-2

Processus de validation des recommandations du W3C

Chaque document publié comporte une section « statut » mentionnant l'étape du processus de validation.

Pour satisfaire les intérêts stratégiques d'interopérabilité et d'adaptation à la fois culturelle et linguistique, le W3C a soumis les WCAG 2 à l'organisme international de standardisation ISO/IEC.

Au sein de l'organisation ISO, le comité technique JTC 1 réunit des experts qui cherchent à développer ensemble des standards mondiaux concernant les technologies de l'information et de la communication pour des applications visant les entreprises et les consommateurs. Pour devenir une « spécification disponible publiquement », aussi appelée PAS (*Publicly Available Specification*), les WCAG 2 ont dû s'inscrire dans le processus de l'ISO. Cela consiste à transposer la procédure au sein de laquelle les organisations accréditées comme des soumissionnaires PAS validés peuvent soumettre leur spécification directement au vote des pays, pour devenir un standard ISO/IEC. Fin octobre 2010, après trois mois de délibérations par les membres nationaux du comité JTC 1 de l'ISO/IEC, le W3C a été admis en tant que soumissionnaire PAS reconnu pour l'ensemble des documents définis comme « toute technologie web fondamentale stable produite par le W3C, qui se trouvent aussi faire partie du JTC 1 ».

Vous trouverez plus d'informations à ce sujet sur le site du W3C (en anglais) :

► <http://ur1.ca/a40pd^a>

a. <http://www.w3.org/2010/04/pasfaq.html>

Les quatre grands principes des WCAG 2

Les WCAG représentent les règles d'accessibilité relatives aux contenus web. Par « contenus », la WAI entend tout type de contenu restitué par le navigateur web : textes, images, vidéos, enregistrements audio, documents en téléchargement, animations, jeux interactifs... Bref, *tout ce qui apparaît à l'écran quand vous saisissez l'adresse d'une page web* est considéré comme du contenu et se doit d'être accessible.

Les WCAG 2 sont réparties selon quatre grands principes :

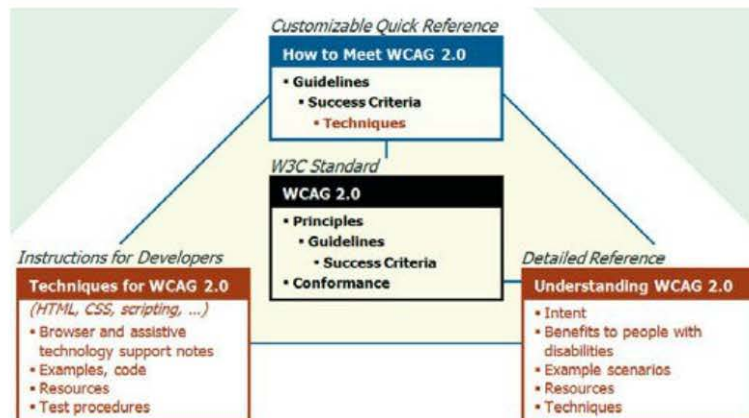
- 1 Perceptible** : les contenus doivent pouvoir être perçus par tout utilisateur, c'est-à-dire qu'ils doivent être visibles, audibles et restituables pour ceux ne pouvant pas les voir ou les entendre.
- 2 Utilisable** : les contenus doivent pouvoir être utilisés quelle que soit le type d'utilisation de l'ordinateur : utilisation exclusive du clavier, d'un clavier virtuel, d'un pointeur laser ou toute autre technologie d'assistance.
- 3 Compréhensible** : les contenus doivent être lisibles. L'internaute doit pouvoir naviguer de façon logique, être guidé dans sa navigation et accompagné s'il se perd ou fait une erreur sur un formulaire, par exemple.
- 4 Robuste** : le code produit doit respecter les standards de façon à permettre la compatibilité avec les différents supports existants ou à venir.

Ces grands principes se veulent intangibles, intemporels et indépendants de tout contexte technologique. Quels que soient le type de contenu et la forme qu'il prend, s'il est perceptible, utilisable, compréhensible et robuste, le contenu sera dit accessible.

Douze grandes règles issues des quatre principes

Les règles, elles-mêmes déclinées en critères de succès et vérifiables grâce à différentes techniques, sont la partie opérationnelle des WCAG 2. Vous les trouverez traduites en français sur le site du W3C : <http://www.w3.org/Translations/WCAG20-fr/>.

Figure 4-3
Le standard WCAG 2
et les autres documents



On compte douze grandes règles, chacune s'inscrivant dans l'un des quatre grands principes des WCAG 2 :

- Principe 1 : perceptible
 - **Règle 1.1** : proposer des équivalents textuels à tout contenu non textuel qui pourra alors être présenté sous d'autres formes selon les besoins de l'utilisateur : grands caractères, Braille, synthèse vocale, symboles ou langage simplifié.
 - **Règle 1.2** : proposer des versions de remplacement aux médias temporels².
 - **Règle 1.3** : créer un contenu qui puisse être présenté de différentes manières sans perte d'information ni de structure (par exemple, avec une mise en page simplifiée).
 - **Règle 1.4** : faciliter la perception visuelle et auditive du contenu par l'utilisateur, notamment en séparant le premier plan de l'arrière-plan.
- Principe 2 : utilisable
 - **Règle 2.1** : rendre toutes les fonctionnalités accessibles au clavier.
 - **Règle 2.2** : laisser à l'utilisateur suffisamment de temps pour lire et utiliser le contenu.
 - **Règle 2.3** : ne pas concevoir de contenu susceptible de provoquer des crises d'épilepsie.
 - **Règle 2.4** : fournir à l'utilisateur des éléments d'orientation pour naviguer, trouver le contenu et se situer dans le site.
- Principe 3 : compréhensible
 - **Règle 3.1** : rendre le contenu textuel lisible et compréhensible.
 - **Règle 3.2** : faire en sorte que les pages apparaissent et fonctionnent de manière prévisible.
 - **Règle 3.3** : aider l'utilisateur à éviter et à corriger les erreurs de saisie.
- Principe 4 : robuste
 - **Règle 4.1** : optimiser la compatibilité avec les agents utilisateurs actuels et futurs, y compris avec les technologies d'assistance.

Ces grandes règles sont encore très générales et ne sont pas d'une grande utilité lorsqu'on souhaite savoir comment faire concrètement pour « ne pas concevoir de contenu susceptible de provoquer des crises d'épilepsie » (règle 2.3), par exemple.

À chaque règle sont donc associés un ou plusieurs critères de succès.

2. Voir le chapitre 7, consacré à la thématique « Multimédia » d'AccessiWeb.

Les critères de succès

Les critères de succès indiquent l'*objectif opérationnel à atteindre* pour respecter la règle. Ils sont normatifs, c'est-à-dire que s'ils ne sont pas respectés, la règle ne pourra pas être validée et le contenu ne sera pas accessible selon le niveau d'accessibilité visé.

Pour le cas de la règle 2.3 prise en exemple, dans le principe « Utilisable », on considère deux critères de succès :

- **Critère 2.3.1** : pas plus de trois flashes ou sous le seuil critique. Une page web doit être exempte de tout élément qui flashe plus de trois fois dans n'importe quel intervalle d'une seconde, ou ce flash doit se situer sous le seuil de flash générique et le seuil de flash rouge (niveau A).
- **Critère 2.3.2** : trois flashes. Une page web doit être exempte de tout élément qui flashe plus de trois fois dans n'importe quel intervalle d'une seconde (niveau AAA).

SOURCE La règle 2.3 et ses critères de succès

► <http://ur1.ca/9wp96^a>

a. <http://www.w3.org/Translations/WCAG20-fr/#seizure>

Les deux critères de succès de la règle 2.3 sont très proches, mais le niveau d'exigence est plus important pour le critère 2.3.2 qui ne supporte aucune exception, alors que le critère 2.3.1 accepte plus de trois flashes s'ils sont sous le seuil critique. Nous verrons ci-après que cela tient aux différents niveaux de conformité.

EN PRATIQUE Critères WCAG 2 : mais concrètement, comment fait-on ?

Vous pouvez observer que les critères de succès des WCAG 2 restent assez... abscons. Mais rassurez-vous, nous vous présenterons dans le chapitre suivant un référentiel qui vulgarise WCAG 2 en le rendant plus simple. L'objet de ce chapitre est de vous montrer la structure des WCAG 2 et leur mode de fonctionnement général, car c'est des WCAG 2 que s'inspirent aussi bien AccessiWeb 2.2 que le RGAA 2.2.1.

Pour valider un critère de succès, les WCAG 2 n'imposent aucune technologie ni aucune technique. Il y a donc une grande liberté pour arriver au résultat. Mais afin d'aider les développeurs, la WAI a complété son dispositif en associant des techniques aux critères de succès.

Les techniques et échecs WCAG

On compte trois sortes de techniques :

- Les **techniques suffisantes** (*Sufficient Techniques*) : il suffit d'appliquer la technique proposée pour valider le critère de succès. Il s'agit de techniques minimales pour valider la conformité au critère.
- Les **techniques à titre indicatif** (*Advisory Techniques*) : il s'agit de techniques recommandées permettant d'améliorer la validation du critère de succès. Cependant, elles ne sont pas toujours suffisantes pour valider le critère à elles seules, ni obligatoirement pertinentes dans tous les cas.
- Les **échecs** (*Failures*) : il s'agit de contre-exemples. En appliquant la technique décrite, on est certain d'invalidier le critère de succès.

Pour obtenir la liste des techniques associées à un critère, sur la page des WCAG traduites en français³, il suffit de cliquer sur le lien « Comment satisfaire à [numéro du critère] » situé à droite du critère de succès. Toutes les techniques et échecs sont recensés dans un document intitulé *How to Meet WCAG 2*, que l'on pourrait traduire par « Comment respecter WCAG 2 ? », disponible en ligne à l'adresse :

<http://www.w3.org/WAI/WCAG20/quickref/>

Les techniques sont numérotées en fonction de la technologie à laquelle elles se réfèrent.

On compte douze techniques différentes :

- Les techniques préfixées de la lettre G correspondent aux **techniques générales**, ne se rapportant pas à une technologie en particulier et ne s'appliquant que si la technologie le permet. Par exemple, la technique G1 indique qu'il faut ajouter un lien en début de chaque page dirigeant vers la zone de contenu principal.
- Les techniques préfixées de la lettre H se rapportent à des **techniques HTML et XHTML**. Par exemple, la technique H4 indique la création d'un ordre de tabulation logique à travers les liens, les contrôles de formulaire et les objets.
- Les techniques préfixées de la lettre C concernent les **techniques CSS**. Par exemple, la technique C18 préconise l'utilisation de `margin` et `padding` CSS en lieu et place des images d'espacement pour la conception de la mise en page.
- Les techniques SCR se rapportent aux **techniques de script côté client**. Par exemple, la technique SCR1 demande à ce qu'il soit permis à l'utilisateur d'augmenter la limite de temps par défaut.
- Les techniques SVR concernent les **techniques de script côté serveur**. Par exemple, la technique SVR2 préconise l'utilisation d'un document `.htaccess` pour s'assurer que la seule façon d'accéder à un contenu non conforme est de le faire à partir d'un contenu conforme.

3. <http://www.w3.org/Translations/WCAG20-fr/>

- Les techniques SM correspondent aux **techniques SMIL**. SMIL est peu connu au sein du Web et peu utilisé, il s'agit pourtant d'une spécification stable du W3C signifiant « langage d'intégration multimédia synchronisée ». Par exemple, la technique SM12 indique qu'il faut fournir du sous-titrage pour sourds et malentendants à travers les flux de texte synchronisés en SMIL 2.0.
- Les techniques préfixées de la lettre T concernent les **techniques en texte brut**. Par exemple, la technique T1 indique qu'il faut utiliser les conventions de formatage de texte pour les paragraphes.
- Les **techniques ARIA** sont relatives à la spécification WAI-ARIA. Par exemple, la technique ARIA2 préconise l'identification des champs requis avec la propriété `aria-required`.
- Les **techniques FLASH** sont relatives... à Flash bien entendu. Par exemple, la technique FLASH16 indique qu'il faut rendre accessible les actions au clavier en utilisant l'événement `click` sur les composants standards.
- Les techniques SL sont relatives à **Silverlight**. Par exemple, la technique SL5 demande à ce que soit définie une classe image `tabulable` pour Silverlight.
- Les **techniques PDF** sont, comme il faut s'y attendre, relatives au format PDF. Par exemple, la technique PDF2 indique qu'il faut créer des repères (*bookmarks*) dans les documents PDF.
- Les techniques préfixées de la lettre F correspondent aux **techniques fréquentes qui invalident** tel ou tel critère. La lettre F fait référence à l'anglais *Failure*, « échec » en français. Par exemple, la technique F3 s'intitule « Échec au critère de succès 1.1.1 dû à l'utilisation de CSS pour insérer des images qui véhiculent une information importante ».

SOURCE Les différentes catégories de techniques WCAG 2

► <http://ur1.ca/9wp9a>^a

a. <http://www.w3.org/TR/2012/NOTE-WCAG20-TECHS-20120103/intro.html#intro-tech-types>

Insistons sur le fait que *ces techniques ne sont pas obligatoires* pour valider un critère de succès. Le choix est laissé aux développeurs : soit ils implémentent au moins une technique suffisante, soit ils fournissent un moyen alternatif dont ils ont vérifié la compatibilité avec l'accessibilité, ce qui est en pratique très contraignant. Il s'agit donc d'outils donnant des exemples et contre-exemples de ce qu'il est possible de faire pour valider un critère. Ces techniques ne sont pas figées, elles sont amenées à s'enrichir au fur et à mesure des évolutions technologiques.

Seule la validation des critères de succès est impérative pour respecter les WCAG.

Notons également que les principes, règles et critères de succès, qui représentent le volet obligatoire pour être conforme à WCAG 2 et pouvoir prétendre être accessible selon le W3C, sont *neutres techniquement*. Autrement dit, les règles et critères de succès ne reposent sur aucune technologie et n'imposent aucune technique pour les respecter.

L'objectif du W3C était clairement de parer à l'échec des WCAG 1 qui, en s'appuyant sur des techniques parfois dépassées et en ne parvenant pas à prendre en compte les évolutions techniques du Web, avaient tendance à donner une image de l'accessibilité web figée, arriérée et interdisant toutes les avancées du Web. Il fallut en effet 10 ans pour passer des WCAG 1 aux WCAG 2. Le processus étant long et fastidieux, il était nécessaire de trouver un mode de mise à jour plus réactif face aux avancées techniques.

En s'affranchissant des technologies et en précisant que les principes, règles et critères de succès sont valables quelle que soit la technologie, actuelle ou à venir, la WAI a voulu rendre son dispositif WCAG 2 plus évolutif.

Les techniques, qui ne sont pas normatives et peuvent évoluer de façon souple et ouverte grâce à l'apport de chacun, viennent astucieusement compléter ce dispositif pour l'ancrer dans la réalité du Web d'aujourd'hui, que celui-ci se situe en 2012 ou en 2022.

RESSOURCES Les WCAG 2 en un clin d'œil

La WAI publie sur son site de la documentation afin de faciliter la diffusion de l'accessibilité du Web. Bien qu'il ne soit pas toujours aisé de trouver ces informations, et que leur consultation ne soit pas toujours agréable visuellement, elles sont souvent de qualité et méritent que l'on s'y intéresse.

L'un de ces documents, intitulé « *WCAG 2 at a glance* » offre une synthèse simplifiée des règles réparties selon les quatre grands principes évoqués précédemment. En voici une traduction :

- Perceptible
 - Fournir des alternatives textuelles pour les contenus non textuels.
 - Fournir du sous-titrage et autres alternatives pour le multimédia.
 - Créer du contenu qui puisse être présenté de différentes façons, y compris par les technologies d'assistance, sans perte de sens.
 - Faciliter la visualisation et l'écoute des contenus.
- Utilisable
 - Permettre l'accès au clavier de toutes les fonctionnalités.
 - Donner suffisamment de temps aux utilisateurs pour lire et utiliser les contenus.
 - Ne pas utiliser de contenus provoquant des crises d'épilepsie.
 - Aider les utilisateurs à naviguer et à trouver les contenus.
- Compréhensible
 - Rendre le texte lisible et compréhensible.
 - Permettre une apparition et une utilisation prévisible des contenus.
 - Aider les utilisateurs à prévenir et à corriger les erreurs.
- Robuste
 - Optimiser la compatibilité avec les outils utilisateurs actuels et à venir.

Document du W3C sous licence libre de reproduction :

► <http://ur1.ca/9wp9e^a>

Lien vers le document original en anglais :

► <http://ur1.ca/9wp9i^b>

a. <http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>

b. <http://www.w3.org/WAI/WCAG20/glance/>

Les trois niveaux de conformité : A, AA et AAA

Vous vous êtes efforcé de lire et de respecter au mieux les WCAG 2. Votre site, qui auparavant n'était pas accessible, le devient et vous aimeriez pouvoir valoriser vos efforts en clamant haut et fort sur votre site qu'il respecte les WCAG 2, avec un petit logo, par exemple. Or, pour ce faire, la WAI a mis en place des règles à respecter (encore des règles, oui...). Car si *vous n'êtes pas obligé de faire une déclaration de conformité aux WCAG 2 pour suivre leurs préconisations et améliorer le niveau d'accessibilité de vos pages web*, vous ne pouvez pas affirmer respecter les WCAG 2 sans effectuer une déclaration de conformité, qui doit comporter certaines mentions obligatoires.

Figure 4-4

Exemple de logos de conformité aux WCAG 2 selon le niveau A ou AA, disponibles sur le site de la WAI



RESSOURCES Logos de conformité aux WCAG 2

Des logos sur fond jaune ou bleu sont disponibles sur le site de la WAI pour indiquer qu'une page web respecte le niveau A, AA ou AAA des WCAG 2.

► <http://ur1.ca/9wp9i^a>

a. <http://www.w3.org/WAI/WCAG2-Conformance>

Le dispositif WCAG 2.0 est structuré en deux grandes sections principales : une section sur les règles WCAG 2.0 et une section relative à la conformité.

Être conforme à un standard signifie satisfaire aux exigences préconisées par ce standard. Concernant les WCAG 2, l'exigence est de respecter les critères de succès. Ces derniers sont déclinés en trois niveaux d'accessibilité : le niveau A (niveau minimal de conformité), le niveau AA (niveau intermédiaire) et le niveau AAA (niveau d'amélioration).

Cela peut paraître étonnant, mais la WAI indique clairement dans une note qu'il ne faut pas chercher à atteindre le niveau AAA :

« Il n'est pas recommandé de se fixer le niveau AAA comme objectif à l'échelle de sites entiers car il n'est pas possible de satisfaire à tous les critères de succès du niveau AAA pour certains contenus. »

Le niveau accessible selon les WCAG 2 est donc le niveau AA.

DÉBAT Pertinence des trois niveaux WCAG 2

Alors que seul le niveau AAA est considéré comme un niveau d'amélioration, la question de l'implémentation de tous les critères A et AA dans un contexte réel est débattu. Est-ce possible, et surtout, est-ce bien pertinent ?

Nous aborderons cette question plus en détail dans le chapitre 13 traitant de la démarche accessibilité.

Satisfaire à un niveau de conformité (ce qui vous donne le droit d'afficher le petit logo du niveau atteint) signifie respecter tous les critères de succès du niveau concerné. La vérification du niveau de conformité ne peut se faire que sur une page complète. On ne peut pas dire être accessible sur une partie de page seulement. Dans le cas d'un processus qui lie plusieurs pages entre elles (la commande en ligne d'un article ou la création d'un compte utilisateur, par exemple), c'est tout le processus qui doit être vérifié, l'ensemble des pages constituant une unité fonctionnelle.

Pour afficher le logo de conformité WCAG 2 de niveau AA, il faut avoir respecté tous les critères de succès de niveau A et de niveau AA, les niveaux étant cumulatifs.

COMPRENDRE Version de remplacement conforme

Si l'une de vos pages web contient un élément qu'il est impossible de rendre accessible nativement, les WCAG autorisent l'utilisation d'une version de remplacement conforme.

Pour cela, il faut que la version de remplacement :

- respecte tous les critères de succès du niveau que vous prétendez atteindre ;
- fournisse les mêmes informations et des fonctionnalités similaires à celles de la version non accessible ;
- soit aussi à jour que la version non conforme.

Par ailleurs, il faut pouvoir accéder à cette version conforme de façon accessible, soit en faisant de cette version conforme la version par défaut, soit en prévoyant un mécanisme accessible permettant d'atteindre cette version conforme.

► <http://ur1.ca/9wp9n^a>

a. <http://www.w3.org/Translations/WCAG20-fr/#conforming-alternate-versiondef>

L'objectif de l'affichage de ces logos, outre la valorisation de vos efforts par une action de communication, concerne avant tout la *valorisation de l'accessibilité web* elle-même, et l'information aux personnes en situation de handicap. C'est cette information qu'il est demandé de fournir au sein de la déclaration de conformité.

Vous souvenez-vous de la partie concernant le RGAA et son attestation de conformité (voir le chapitre 3) ? Comme indiqué précédemment, les WCAG 2 constituent la référence et le dispositif français du RGAA en découle. Il en va de même pour la déclaration de conformité, ou attestation de conformité selon le RGAA.

Il s'agit d'une autodéclaration indiquant précisément le niveau d'accessibilité respecté. À noter qu'il est recommandé d'indiquer également les critères de succès respectés au-delà du niveau atteint. Ainsi, si votre page web ne respecte que le niveau A, mais que vous respectez par ailleurs plusieurs critères de niveau AA et AAA, vous êtes encouragé à l'indiquer.

Notons par ailleurs qu'on ne parle de conformité que pour des pages web. Si vous estimez que votre site entier est conforme, vous devez indiquer précisément les pages sur lesquelles vous avez basé vos tests.

Comme pour le RGAA, les mentions requises pour la déclaration de conformité à WCAG 2 sont :

- la date de la déclaration ;
- l'intitulé des règles, leur version et leur URI : « Règles pour l'accessibilité des contenus Web 2.0 à l'adresse <http://www.w3.org/TR/2008/REC-WCAG20-20081211/> » ;
- le niveau de conformité satisfait (niveau A, AA ou AAA) ;
- une brève description des pages web, comme une liste d'URI, pour lesquelles la déclaration est effectuée en précisant si les sous-domaines sont inclus dans la déclaration ;
- une liste des technologies web dont dépend l'utilisation du contenu.

SOURCE La conformité aux WCAG 2

Pour plus de détails sur la conformité aux WCAG 2, consultez la section dédiée des WCAG 2 (en français) à l'adresse suivante :

► <http://ur1.ca/9wp9z^a>

a. <http://www.w3.org/Translations/WCAG20-fr/#conformance>

DÉBAT « Compatible avec l'accessibilité »

Nous avons vu que les principes, règles et critères de succès des WCAG 2 ont été élaborés en dehors de toute considération technologique. Cette volonté affichée de rendre les WCAG 2 neutres technologiquement vise à permettre une validité des WCAG 2 quelles que soient les évolutions du Web.

Or, cette neutralité est compensée par une notion qui pour le coup est très ancrée dans la pratique technique des utilisateurs : la compatibilité avec l'accessibilité. Les contenus web doivent donc être, selon les WCAG 2, « compatibles avec l'accessibilité », c'est-à-dire « compatibles avec les technologies d'assistance des utilisateurs ainsi qu'avec les fonctions d'accessibilité des navigateurs et des autres agents utilisateurs ». Mais libre à chacun de réaliser ses propres tests, la WAI refusant de donner un avis qui risquerait de faire interdire telle ou telle technologie.

On voit là la limite des WCAG 2 qui donnent des principes généraux, avec 437 techniques diverses et variées pour savoir comment faire, mais se refusent à faire l'état des lieux qui s'impose pour déterminer ce qui est ou non compatible avec l'accessibilité à un temps *t*.

Cette complexité à déterminer ce qui est compatible avec l'accessibilité est bien retranscrite dans la définition de « compatible avec l'accessibilité » que donnent les WCAG :

► <http://ur1.ca/9wpa7a>

Nous verrons dans le chapitre suivant une méthode d'évaluation des WCAG 2, AccessiWeb 2.2, qui prend le parti de répondre à la question de façon plus claire, quitte à qualifier certaines technologies d'inaccessibles en leur imposant systématiquement une alternative.

a. <http://www.w3.org/Translations/WCAG20-fr/#accessibility-supporteddef>

Évaluer l'accessibilité d'un site web

Pour savoir si un site est accessible, il faut pouvoir déterminer s'il respecte les critères de succès des WCAG 2 et donc vérifier que ces critères sont validés. Il convient donc de réaliser une *évaluation* des critères d'accessibilité, aussi appelée « audit d'accessibilité web ».

Principe de l'audit

Voici la définition d'un audit selon Wikipédia⁴ :

« L'audit est une activité de contrôle et de conseil qui consiste en une expertise par un agent compétent et impartial et un jugement sur l'organisation, la procédure, ou une opération quelconque de l'entité.

4. <http://fr.wikipedia.org/wiki/Audit>

L'audit est surtout un outil d'amélioration continue, car il permet de faire le point sur l'existant (état des lieux) afin d'en dégager les points faibles et/ou non conformes (suivant les référentiels d'audit).

Cela, afin de mener par la suite les actions adéquates qui permettront de corriger les écarts et dysfonctionnements constatés. »

Un audit, quel que soit le domaine, est donc une évaluation de l'existant à un temps *t* selon les normes d'un référentiel. Dans le cas d'un audit d'accessibilité, il s'agit de vérifier, au sein d'une page web, que tous les critères du référentiel choisi (WCAG 2, AccessiWeb 2.2 ou RGAA 2.2.1, par exemple) sont respectés.

Concrètement, on vérifie critère par critère si le référentiel est respecté. Trois résultats sont possibles.

- Le critère est **validé** : aucune erreur n'a été trouvée.
- Le critère est **invalidé** : une seule erreur suffit pour invalider le critère. Ainsi, si votre page web contient de nombreuses images et qu'elles comportent toutes un attribut `alt` sauf une, le critère demandant à ce que les images aient un attribut `alt` sera invalidé à 100 %, même si l'erreur ne se produit que dans 1 % des cas.
- Le critère est **non applicable** : par exemple, votre page web ne contient pas de formulaire. Tous les critères concernant les formulaires seront non applicables.

Le travail d'audit pourrait donc être assimilé à celui d'un inspecteur à la recherche d'erreurs.

IDÉE REÇUE **Mon site est accessible à 78 % !**

Quand on rédige un rapport d'évaluation, la tentation est grande d'ajouter de jolis schémas représentant le pourcentage d'accessibilité des pages testées. Or, comme nous l'avons vu, il suffit d'une erreur pour invalider un critère. Ainsi, qu'il s'agisse d'une erreur bénigne, triviale à corriger, ou de 150 erreurs demandant un gros travail de correction, le résultat de l'audit sera le même : le critère sera invalidé. Si les chiffres, les graphiques et autres schémas sont autant d'outils utiles à la restitution d'un audit, il convient donc de bien situer le contexte. Une erreur sur un critère ne signifie pas grand-chose en termes de travail à effectuer pour la corriger. Il est important de toujours le préciser.

De même, si 70 % des critères sont validés pour une page, cela ne signifiera pas que cette dernière sera accessible à 70 %. En effet, tous les critères n'ont pas la même importance en termes d'accès à l'information, et le découpage en trois niveaux A, AA et AAA est insuffisant pour se situer.

Votre page web peut même valider tous les critères, sauf un, et être inaccessible à 100 % pour certaines personnes. Prenons l'exemple d'une page contenant un piège au clavier au début du code source, empêchant la navigation au clavier : toute personne ne pouvant utiliser un autre mode de navigation que le clavier ne pourra jamais accéder à vos contenus, et ce, même si vous respectez par ailleurs tous les autres critères d'accessibilité.

Cela signifie que les critères d'accessibilité en tant que tels, malgré leur déclinaison en trois niveaux de conformité, ne peuvent servir d'unité de mesure pour évaluer l'accessibilité d'une page web. Or, avoir une unité de mesure objective est une nécessité, surtout quand on souhaite faire de l'accessibilité sur une multitude de pages, à une échelle industrielle.

Des groupes de travail se sont formés pour étudier cette question et tenter d'y répondre en proposant de nouveaux moyens pour mesurer l'accessibilité à partir des WCAG 2. C'est le cas de MIPAW, la méthode d'intégration progressive d'accessibilité du Web coordonnée par BrailleNet (voir le chapitre 13).

En attendant d'avoir de nouvelles unités de mesure plus pertinentes, prenez donc les référentiels d'accessibilité pour ce qu'ils sont, à savoir des outils à utiliser avec discernement, ce que nous essaierons de vous aider à faire avec cet ouvrage.

Si les WCAG 2 font office de référence, nous avons vu précédemment qu'elles sont difficilement utilisables en l'état. Plusieurs raisons peuvent être évoquées :

- la rédaction et la structure de consultation ne sont pas simples ;
- une partie importante des ressources (les techniques) n'est pas traduite ;
- les WCAG 2 se situent en dehors de tout contexte technologique, laissant aux développeurs et évaluateurs le soin de juger si la pratique est « compatible avec l'accessibilité ».

Des méthodes d'évaluation des WCAG 2 ont donc été élaborées afin de vérifier plus facilement si les critères de succès des WCAG 2 sont bien respectés.

Méthodes d'évaluation

En France, les différents référentiels d'accessibilité partagent un même objectif commun : évaluer la conformité aux WCAG 2. Deux méthodes d'évaluation ont été élaborées.

À SAVOIR Méthode d'évaluation ou méthode d'application ?

S'il n'est pas rare que les règles d'accessibilité et les référentiels servent pour concevoir et créer des sites web accessibles, l'objectif commun d'AccessiWeb et du RGAA est avant tout de vérifier la conformité d'une page avec les WCAG et d'évaluer le niveau de respect de ces règles. Ces référentiels ne disent pas comment respecter la règle, même si l'on peut parfois le déduire, ils permettent de savoir si elle est validée ou non. Pour reprendre une métaphore d'Aurélien Levy, expert accessibilité chez Temesis, qui prolonge notre analogie avec le bâtiment, nous pouvons comparer cela au fait de planter un clou.

La règle serait : un clou doit être bien planté.

Méthode d'application : prendre son marteau d'une main, le clou dans l'autre, taper sur la tête du clou, au centre, en le maintenant bien droit et en donnant des coups secs et réguliers.

Méthode d'évaluation :

- La personne tient-elle bien son marteau d'une main et le clou dans l'autre ?
- Le clou est-il bien droit ?
- La personne tape-t-elle bien sur la tête du clou, au centre ?
- Le rythme de frappe est-il régulier ?

Le référentiel AccessiWeb

Historique d'AccessiWeb

L'association BrailleNet est un acteur historique de l'accessibilité numérique en France. Dès 2002, elle lance son initiative AccessiWeb, marque déposée par l'association visant à promouvoir l'accessibilité du Web à travers les règles internationales WCAG. Le référentiel AccessiWeb⁵ voit le jour en 2003, accompagné du label AccessiWeb.

À SAVOIR BrailleNet

Créée en 1998, BrailleNet est une association loi 1901 à but non lucratif visant à promouvoir l'accessibilité numérique, en particulier pour les personnes présentant un handicap visuel. Initiatrice du projet AccessiWeb, et maintenant de l'Institut de l'accessibilité numérique (IAN), BrailleNet a élargi son champ d'action à toutes les situations de handicap, en promouvant les règles d'accessibilité du W3C, dont elle est le coordinateur officiel pour la traduction française autorisée.

Pour en savoir plus sur BrailleNet, consultez le site web de l'association :

► <http://www.brailenet.org>

Et pour de plus amples renseignements sur l'IAN, rendez-vous à l'adresse suivante :

► <http://www.accessibilite-numerique.org>

S'appuyant tout d'abord sur son réseau restreint d'experts, l'association met en place des formations pour pallier le manque d'expertise en matière d'accessibilité du Web et créer une communauté de compétences, nommée GTA pour Groupe de travail AccessiWeb. Les membres du GTA ont suivi la formation d'« experts AccessiWeb en évaluation » proposée par BrailleNet et ont obtenu leur examen final validant leurs acquis.

5. <http://www.accessiweb.org/index.php/referentiels.html>

DÉBAT Experts, vous avez dit experts ?

Les formations à l'accessibilité proposées par BrailleNet ont été parmi les premières en France, permettant l'émergence d'une communauté d'experts regroupés au sein du groupe de travail AccessiWeb (GTA). Le GTA constitue une communauté professionnelle française de l'accessibilité du Web, sans autre véritable équivalent. Disposant d'une liste de discussion privée dont l'accès est réservé à ses membres, et de rencontres semestrielles pour faire le point sur les avancées techniques en la matière (les séminaires AccessiWeb), précieux outils dans la pratique de l'accessibilité web au quotidien, le GTA gagnerait toutefois à s'élargir. Une réflexion est en cours, engagée par BrailleNet, pour ouvrir le GTA sous certaines conditions, et le doter par exemple d'un wiki ouvert à tous qui permettrait de capitaliser sur les retours d'expériences échangés sur la liste de discussion.

Cette réflexion porte également sur la revalorisation de l'expression « expert AccessiWeb en évaluation » qui, pour le moment, signifie uniquement que les personnes ont suivi la formation de BrailleNet et réussi l'examen à un temps *t*, mais pas qu'ils pratiquent l'accessibilité du Web régulièrement ni que leurs connaissances en la matière sont à jour. À suivre donc...

Parallèlement, l'offre de formation s'est développée et diversifiée, offrant différentes approches pédagogiques dispensées par d'autres acteurs que BrailleNet. Ainsi, l'expertise en accessibilité n'est pas confinée au sein du GTA. Les personnes compétentes sont nombreuses, formées à l'accessibilité du Web par d'autres voies, que ce soit à l'université, dans d'autres centres de formation privés, en autodidacte (plus rare !) ou sur d'autres référentiels.

Si vous désirez discuter avec d'autres personnes sur l'accessibilité du Web et vous former, en complément de la lecture de ce livre, vous pouvez par exemple échanger sur le forum dédié à l'accessibilité du site Alsacréations^a, vous inscrire à la liste de discussion francophone AccessTech^b ou encore venir rencontrer de nombreux experts à Paris Web^c ou lors de l'AccessCamp^d de l'April. Inutile d'être expert pour parler d'accessibilité web !

a. <http://forum.alsacreations.com/list-6-Accessibilite-du-Web.html>

b. <http://listes.rezo.net/mailman/listinfo/accesstech>

c. <http://www.paris-web.fr> : conférence sur le Web design, la qualité et l'accessibilité web ayant lieu à Paris, chaque année au mois d'octobre.

d. <http://accesscamp.april.org> : barcamp sur l'accessibilité et les logiciels libres ayant lieu chaque année fin janvier.

Bien que la rédaction du référentiel AccessiWeb soit coordonnée par BrailleNet, elle est le fruit d'un travail collectif mené par les membres du GTA.

Une première version du référentiel est intégralement rédigée par un expert (en l'occurrence Jean-Pierre Villain), passée en revue par deux autres experts (Aurélien Levy et Laurent Denis pour le référentiel AccessiWeb 2.2). Ensuite, un groupe restreint d'une vingtaine d'experts référents sélectionnés par BrailleNet participe à une première relecture pour valider les modifications apportées au référentiel, à mesure de son évolution, et produire une version candidate du référentiel, validée par

l'ensemble des acteurs (auteur, groupe d'écriture, experts référents, BrailleNet). Il n'y a pas d'appel à commentaire public.

Ce processus a permis une évolution régulière du référentiel AccessiWeb, corrigeant les éventuelles lacunes au fur et à mesure des versions, et se calquant sur les évolutions des WCAG. Depuis 2003, ce travail collaboratif s'est structuré, apportant ainsi une certaine maturité à la méthodologie de production du référentiel.

Structure du référentiel AccessiWeb

Devant la complexité de lecture des WCAG 2, le référentiel AccessiWeb a pris le parti de réorganiser sa structure pour en rendre la lecture plus opérationnelle, en répartissant les critères de succès selon treize thématiques :

- 1** Images
- 2** Cadres
- 3** Couleurs
- 4** Multimédia
- 5** Tableaux
- 6** Liens
- 7** Scripts
- 8** Éléments obligatoires
- 9** Structuration de l'information
- 10** Présentation de l'information
- 11** Formulaires
- 12** Navigation
- 13** Consultation

Chaque thématique est introduite par une recommandation générale reprenant l'ensemble des exigences qui seront explicitées par des critères.

Par exemple, pour la thématique 1 (Images), la recommandation est la suivante :

« Donner à chaque image porteuse d'information une alternative textuelle pertinente et une description détaillée si nécessaire. Remplacer les images textes par du texte stylé lorsque c'est possible. »

Au sein de chaque thématique, nous retrouvons une liste de critères AccessiWeb rédigés sous forme de questions dont la réponse peut être « oui : critère validé », « non : critère non validé » ou « critère non applicable ». Notons que ces critères respectent la déclinaison en trois niveaux des WCAG 2, mais que ces niveaux s'intitulent différemment : le niveau A est appelé « niveau Bronze », le niveau AA est le « niveau Argent » et le niveau AAA correspond au « niveau Or ».

IDÉE REÇUE And the winner is... l'Or ?

La déclinaison des critères faite par AccessiWeb en Bronze, Argent et Or, donne la fausse impression d'un concours olympique où seule la médaille d'or est synonyme de victoire. Pourtant, comme nous l'avons vu avec les niveaux de conformité des WCAG 2 (voir page 48), sur lesquels sont calqués les niveaux du référentiel AccessiWeb, le niveau AAA qui correspond au niveau Or n'est pas recommandé ! Il ne s'agit donc pas de décrocher l'Or pour être accessible, ce qui n'aurait pas de sens dans le contexte des WCAG 2, mais plutôt de viser l'Argent, seul niveau atteignable dans tous les cas de figure.

Nous verrons par ailleurs dans le chapitre 13 que, dans un souci d'efficacité, il vaut mieux envisager l'accessibilité du Web comme un processus d'amélioration progressive et continue, que comme un concours donnant droit à une médaille.

Ces critères ont une correspondance directe avec les critères de succès des WCAG 2, dont mention est faite pour référence.

Liste générale et liste déployée

Le référentiel AccessiWeb, qui se trouve publié dans sa version 2.2 à l'heure où nous écrivons ces lignes, est disponible en deux versions :

- une version générale, qui contient les 133 critères AccessiWeb ;
- une version déployée, qui est composée des 133 critères AccessiWeb et, pour chacun d'entre eux, une liste de tests permettant d'aider l'évaluateur à vérifier la validité du critère sans rien oublier.

Un glossaire accompagne ces deux listes pour éclairer de façon indispensable les différentes notions abordées dans le référentiel.

Prenons un exemple. Le critère AccessiWeb 1.1, de la thématique 1 relative aux images, est rédigé de la façon suivante dans la version générale :

« Critère 1.1 [Bronze] Chaque image a-t-elle une alternative textuelle ? »

Dans la version déployée, le critère est accompagné des tests suivants :

« Test 1.1.1 : Chaque image (balise `<img />`) a-t-elle un attribut `alt` ?

Test 1.1.2 : Chaque zone (balise `<area>`) d'une image réactive a-t-elle un attribut `alt` ?

Test 1.1.3 : Chaque bouton de formulaire (balise `<input>` avec l'attribut `type="image"`) a-t-il un attribut `alt` ?

Test 1.1.4 : Chaque image applet (balise `<applet>`) a-t-elle un attribut `alt` ? »

Enfin, que ce soit dans la version générale comme dans la version déployée, chaque critère est accompagné d'une liste de références mentionnant la correspondance du critère en question avec son équivalent WCAG 2 et RGAA. En effet, à l'annonce de la publication du RGAA comme référence imposée pour le secteur public, AccessiWeb a réalisé un travail important visant à offrir une correspondance parfaite non seulement entre AccessiWeb 2 et WCAG 2, mais également entre AccessiWeb et RGAA.

SOURCE Référentiel AccessiWeb 2.2

La liste générale des critères AccessiWeb 2.2 (sans les tests) est disponible à l'adresse suivante :

► <http://ur1.ca/9ykw5^a>

La liste déployée des critères AccessiWeb 2.2 (avec les tests) est, quant à elle, disponible à cette adresse :

► <http://ur1.ca/9ykw^b>

a. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_generale.html

b. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html

Le RGAA

Nous avons vu précédemment le cadre légal entourant le RGAA (voir le chapitre 3). Intéressons-nous maintenant dans le détail au référentiel technique contenu dans les annexes 1 et 2.

Annexe 1 : les règles du RGAA

L'annexe 1 reprend la liste des règles du RGAA, réparties selon les grands principes WCAG 2. Son intérêt opérationnel par rapport aux WCAG est très limité puisque cette liste reprend textuellement les règles WCAG 2, avant de les développer et les expliciter. Chaque règle et chaque critère de succès associé sont structurés en cinq sections :

- 1** Niveau (le niveau de conformité aux WCAG 2)
- 2** Objectif du critère
- 3** Explication
- 4** Exemple
- 5** Tests associés, renvoyant aux tests de l'annexe 2

L'annexe 1 n'apporte donc pas grand-chose en termes de méthodologie par rapport aux WCAG 2, à ceci près qu'une tentative d'explication pédagogique est présente.

L'intérêt du RGAA réside donc plutôt dans son annexe 2 contenant les tests unitaires.

Annexe 2 : les tests unitaires du RGAA

Les tests unitaires renvoient aux techniques et échecs WCAG 2 (voir page 64). En effet, dans cette annexe, un effort de structuration de l'information en thématiques

opérationnelles a été effectué, sur le modèle d'AccessiWeb. Mais au lieu de compter treize thématiques, on dénombre les douze chapitres thématiques suivants :

- 1 Cadres
- 2 Couleurs
- 3 Formulaires
- 4 Images
- 5 Multimédia
- 6 Navigation
- 7 Présentation
- 8 Scripts
- 9 Standards
- 10 Structure
- 11 Tableaux
- 12 Textes

Cette ressemblance n'est pas fortuite, bien qu'il ne soit fait mention nulle part du référentiel AccessiWeb dans les documents du RGAA.

Alors qu'en 2003, lors de la publication de la première version d'AccessiWeb, l'Agence pour le développement de l'administration électronique (ADAÉ) s'était rapprochée de BrailleNet en mandant l'association pour qu'elle développe son référentiel et forme des experts en vue d'en faire le référentiel de référence de l'administration, l'ADAÉ a disparu en janvier 2006 au profit de la Direction générale pour la modernisation de l'État (DGME), qui a donc été chargée du dossier.

Et alors que le décret d'application de l'article 47 de la loi de 2005 était en préparation, la DGME a décidé de passer par un appel d'offres pour recruter une société ayant pour mission de rédiger un nouveau référentiel : le RGAA.

À SAVOIR ADAÉ, DGME, DISIC

Si l'ADAÉ a cédé sa place à la DGME en 2006, notons que depuis février 2011, la DGME a perdu ses attributions sur l'informatique d'État au profit de la Direction interministérielle des systèmes d'information et de communication de l'État (DISIC). Logiquement, le RGAA devrait donc changer à nouveau d'administration pour relever de la compétence de la DISIC, mais à notre connaissance, rien n'a été communiqué sur ce point.

Pour en savoir plus à ce sujet, vous pouvez consulter l'article « L'État crée une DSI interministérielle » publié le 22 février 2011 sur le site du Monde informatique :

► <http://ur1.ca/9wrwy^a>

a. <http://www.lemondeinformatique.fr/actualites/lire-l-etat-cree-une-dsi-interministerielle-32966.html>

Ainsi est paru en 2009 un nouveau référentiel alors qu'il en existait déjà un bénéficiant d'un retour d'expériences depuis 2003. Compte tenu que la société ayant remporté l'appel d'offres est composée d'experts AccessiWeb membres du GTA, on comprend les ressemblances existant dans la structuration en chapitres thématiques.

La DGME a donc réinventé la roue en somme. Il aurait été plus simple, direz-vous, de garder AccessiWeb comme référentiel de référence, non ? Les raisons de cette décision sont obscures. Une chose est sûre, ce n'est pas par souci d'efficacité qu'elle a été prise. Mais cela permet toutefois d'avoir un référentiel différent, avec une méthodologie propre, ce qui offre un choix.

Voyons donc comment est structurée cette annexe 2 du RGAA.

Composition d'une fiche « test unitaire »

Chaque chapitre thématique comporte des fiches « tests unitaires ». Chaque fiche est structurée en huit sections :

- **Champ d'application** : cette section énumère les composants auxquels le test est applicable.
- **Procédure de test** : au lieu d'avoir une question à laquelle on ne peut répondre que par oui, non ou non applicable (comme dans le référentiel AccessiWeb), le RGAA décrit la procédure de test de façon conditionnelle. Prenons l'exemple du test « 1.1 [Cadres] 1 : Absence de cadres non titrés ». La procédure de test est alors la suivante :
 - « 1. Si l'un des éléments mentionnés dans le champ d'application est présent dans la page, poursuivre le test, sinon le test est non applicable.
 - 2. Si l'élément contient un attribut `title` non vide, le test est validé, sinon le test est invalidé. »
- **Difficulté du test** : trois niveaux sont prévus (facile, moyen et difficile).
- **Techniques WCAG 2** correspondant au test.
- **Failures WCAG 2** (échecs) le cas échéant.
- **Critères de succès WCAG 2**. Notons que ce point et les deux précédents équivalent aux correspondances WCAG 2 dans le référentiel AccessiWeb 2.
- **Niveau WCAG déduit**. Pourquoi « déduit » ? Parce que certains tests peuvent correspondre à différents critères WCAG dont le niveau de conformité est différent. Ainsi, si pour un même test unitaire RGAA on trouve en correspondance un critère WCAG 2 de niveau A et un critère WCAG 2 de niveau AA, le niveau WCAG déduit pour le test sera le niveau le plus bas, soit le niveau A.
- **Profil des intervenants** : quatre profils ont été définis (1. rédacteur et contributeur, 2. graphiste et ergonomes, 3. développeur et intégrateur, 4. référenceur ou communicant). Chaque fiche de test indique donc quel professionnel est davantage concerné par l'application du test.

Voyons maintenant en détail ce qui différencie le référentiel AccessiWeb du RGAA.

Différences entre le référentiel AccessiWeb et le RGAA

Le bref aperçu que nous venons de dresser de ces deux référentiels montre qu'il existe des différences à plusieurs niveaux.

Rédaction, méthodologie et structure

La première différence tient sans doute à la méthodologie d'évaluation supposée par la rédaction même de chacun des référentiels.

Le référentiel AccessiWeb a pris le parti de rédiger ses critères sous forme de question. À chaque critère, une seule question n'appelant qu'une seule et unique réponse possible. Ainsi, pour les images, concernant la nécessité d'avoir un attribut `alt` pertinent, nous retrouvons trois critères distincts :

- Le critère 1.1 pose la question de savoir si les images ont un attribut `alt`. Il s'agit d'une question sur la présence de l'attribut.
« Critère 1.1 [Bronze] Chaque image a-t-elle une alternative textuelle ? »
- Les critères 1.2 et 1.3 posent quant à eux la question de savoir, pour les images ayant un attribut `alt`, si ce dernier est pertinent.
« Critère 1.2 [Bronze] Pour chaque image de décoration ayant une alternative textuelle, cette alternative est-elle vide ? »
« Critère 1.3 [Bronze] Pour chaque image porteuse d'information ayant une alternative textuelle, cette alternative est-elle pertinente (hors cas particuliers) ? »

Ce découpage entre critères de « présence » et critères de « pertinence » se retrouve très fréquemment dans le référentiel AccessiWeb, mais aussi dans le RGAA. On voit qu'à chaque critère, une seule question est posée à laquelle la réponse est simple : oui, non ou non applicable.

Le RGAA, quant à lui, fonctionne sur un mode très différent puisque chaque test soulève toute une série de questions avant de pouvoir dire si le test est validé, non validé ou non applicable. On retrouve toutefois cette dichotomie entre tests de présence et tests de pertinence, mais la rédaction même du test rend la procédure plus échelonnée. Ainsi, si on reprend la rédaction du test concernant les images, voici comment RGAA traite cette question :

- Voyons la procédure de test pour le test « 4.1. [Images] 1 : Présence de l'attribut `alt` » :
 1. Si l'un des éléments mentionnés dans le champ d'application est présent dans la page, poursuivre le test, sinon le test est non applicable.

2. Si l'élément n'est pas un captcha ou ne fait pas partie d'un test qui deviendrait inutile si l'alternative textuelle était présente, poursuivre le test, sinon le test est non applicable.
3. Si l'élément possède un attribut `alt`, le test est validé, sinon il est invalidé⁶.

La procédure d'évaluation via le RGAA est donc progressive, faisant passer l'évaluateur par plusieurs étapes avant de pouvoir donner une réponse. Mais on voit également que la rédaction même du RGAA est plus complexe, avec de très longues phrases utilisant souvent des doubles négations dans la rédaction. Il devient parfois difficile de savoir quelle réponse donner.

Par ailleurs, la structure d'AccessiWeb diffère de celle du RGAA. Alors que le référentiel AccessiWeb 2 est structuré en treize thématiques, RGAA est quant à lui structuré en douze chapitres thématiques n'ayant pas forcément les mêmes intitulés et contenus, ni le même ordre de présentation.

Maintenance et évolutions

Une différence de taille réside dans le processus de rédaction de chaque référentiel ainsi que les maintenances et mises à jour.

Alors que le référentiel AccessiWeb dispose d'une véritable communauté, avec des professionnels faisant remonter leurs retours d'expériences permettant une évolution du référentiel avec correction de bogues et amélioration de la rédaction des critères, tests et définitions du glossaire ; le RGAA est, quant à lui, un document administratif figé.

Il était bien prévu dans le cadre légal la mise en place d'un dispositif permettant une évolution du RGAA, mais les lourdeurs administratives, associées à un manque flagrant de volonté politique, ont depuis sa création condamné le RGAA à être figé. Espérons que nous serons démentis et qu'un dispositif sera mis en place prochainement. Depuis la publication du RGAA en 2009, nous attendons toujours...

Contenus

Concernant les contenus, AccessiWeb et RGAA s'attachent tous deux, chacun à leur façon, à vérifier l'application des WCAG 2. AccessiWeb a de plus réalisé un travail de correspondance entre ses critères et les tests du RGAA. Notons que la correspondance inverse (du RGAA vers AccessiWeb, et des WCAG 2 vers AccessiWeb) n'a pas été travaillée et ne fonctionne pas. Ainsi, si utiliser AccessiWeb 2.2 permet de respecter à la fois les WCAG 2 et le RGAA, l'inverse ne se vérifie pas.

Nous verrons en détail dans le chapitre suivant les différences qui peuvent exister, critère par critère, mais voici déjà un exemple.

6. Voir page 39 de l'annexe 2 sur les tests de conformité au RGAA.

Alors qu'un critère AccessiWeb proscrit l'utilisation du pixel pour définir la taille des caractères (critère 10.4, voir surtout la définition dans le glossaire pour « taille de caractère »⁷), le RGAA ne l'interdit que dans le cas des éléments de formulaire (test 7.14). Ainsi, le niveau d'exigence étant différent, il est possible de respecter le RGAA sans respecter AccessiWeb. Cependant, en respectant AccessiWeb, on est certain de respecter le RGAA.

DÉBAT WCAG et l'utilisation du pixel

La WAI tenant à sa neutralité technologique, elle n'interdit rien sans dire non plus qu'il est possible d'utiliser le pixel. Concrètement, elle a refusé une technique suffisante à ce sujet, tout en refusant également de faire un échec (failure). Elle demande seulement à ce que ce soit « compatible avec l'accessibilité », préférant botter en touche et laisser le soin aux développeurs et évaluateurs de trancher la question. On voit là tout l'intérêt de l'utilisation d'une méthodologie d'évaluation prenant position selon le contexte technologique du moment.

Dans le cas qui nous occupe cependant, AccessiWeb et RGAA ont des orientations différentes... L'accessibilité web n'est pas une matière figée, elle est régulièrement soulevée de débats qui rendent sa pratique d'autant plus passionnante. Mais si vous ne souhaitez pas vous investir dans ces débats, le plus simple est encore de choisir un référentiel et d'en suivre les recommandations.

À SAVOIR De l'évaluation de l'accessibilité à la création d'un site web accessible

Nous avons vu jusqu'ici le principe de l'évaluation de l'accessibilité d'une page web à l'aide d'un référentiel d'accessibilité. Or, peut-être êtes-vous plutôt dans le cas de la création ou de la refonte d'un site web ? Ce livre est-il bien fait pour vous ?

La réponse est : oui, tout à fait !

Comme vous commencez peut-être à le soupçonner, l'accessibilité du Web est une matière à part entière, avec son jargon, ses normes et ses experts. Mais la complexité de la matière ne doit pas vous empêcher de prendre en compte certaines règles d'accessibilité dans votre pratique quotidienne. Tout le monde n'a pas vocation à être expert, et ce livre est avant tout rédigé pour aider les concepteurs de site, quel que soit leur profil, à prendre en compte l'accessibilité dans leur pratique professionnelle.

La première des choses à faire pour vous est donc de comprendre en quoi consiste l'accessibilité du Web et de connaître les règles d'accessibilité. Pour cela, l'ouvrage que vous tenez entre vos mains vous sera d'une aide précieuse, en particulier les chapitres 6 à 11, consacrés aux critères d'accessibilité selon AccessiWeb.

Dans le chapitre 13, nous verrons ensuite comment adopter une démarche en faveur de l'accessibilité.

7. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#mTailleCaractere

Enfin, comme nous l'avons vu maintes fois, aucune technologie n'est recommandée ni proscrite par les WCAG 2. Mais du choix de vos outils et technologies utilisés dépendront grandement le niveau d'accessibilité de votre site. Réfléchissez bien et choisissez donc de préférence des standards ouverts et « compatibles avec l'accessibilité ».

Vous pouvez consulter à ce sujet la FAQ de la WAI répondant à la question « Puis-je respecter les WCAG 2.0 avec JavaScript/Ajax, Flash, PDF et d'autres technologies ? » (en anglais) :

► <http://ur1.ca/9wuv6a>^a

a. <http://www.w3.org/WAI/WCAG20/wcag2faq.html#othertechs>

EN PRATIQUE Automatisation : avantages et limites

Il existe aujourd'hui plusieurs outils permettant de vérifier soit de nombreux critères d'accessibilité d'un seul coup (outils généralistes tels que Tanaguru), soit un critère en particulier, comme l'extension Firefox Color Contrast Checker pour contrôler le contraste des couleurs, ou le validateur du W3C pour vérifier la conformité de votre code aux standards du Web.

Si nous vous encourageons vivement à utiliser ces outils de vérification de façon récurrente pour vous aider à corriger votre code au fur et à mesure de votre travail, nous nous permettons une mise en garde sur le fait qu'aucun outil ne traite tous les critères. L'évaluation de l'accessibilité d'une page web nécessite encore la vérification manuelle de nombreux critères, notamment ceux liés à la pertinence des contenus.

Karl Groves a publié à ce sujet un article présentant une étude qu'il a réalisée sur le nombre de critères automatisables. Il en conclut que seuls 18 % des critères de succès WCAG 2 sont entièrement automatisables ; 31 % peuvent être automatisés mais nécessitent une vérification manuelle, et donc 47 % des critères d'accessibilité ne peuvent être évalués que par un être humain.

Retrouvez cette étude (en anglais) en ligne à l'adresse :

► <http://ur1.ca/a98yx>^a

Ainsi, utilisez les outils automatiques à loisir. Ils sont indispensables pour être efficace, gagner du temps et améliorer votre code de façon progressive. Mais ne négligez surtout pas une évaluation manuelle de l'accessibilité de vos pages web, sans quoi, vous risquez de passer à côté de nombreuses erreurs d'accessibilité, qui représentent autant de freins pour vos utilisateurs...

a. <http://www.karlgroves.com/2012/09/15/accessibility-testing-what-can-be-tested-and-how/>

Maintenant que nous avons vu le fonctionnement et la structure des WCAG 2 et des référentiels AccessiWeb et RGAA, il est temps d'étudier ces règles d'accessibilité en détail. Et c'est à partir du référentiel AccessiWeb 2.2 que nous allons le faire.

Introduction au référentiel AccessiWeb

*« Hâtez-vous lentement, et sans perdre courage,
Vingt fois sur le métier remettez votre ouvrage,
Polissez-le sans cesse, et le repolissez,
Ajoutez quelquefois, et souvent effacez. »*

Nicolas Boileau, L'Art poétique, Chant 1 (1674)



L'objet de ce livre est de vous faire découvrir l'accessibilité du Web en général, au-delà de toute norme particulière. Nous vous avons présenté dans le chapitre précédent les différents standards existant en la matière. Or, pour pouvoir entrer dans le détail, nous avons dû faire un choix et nous avons pris le parti d'étudier, dans les chapitres qui suivent, le référentiel AccessiWeb 2.2.

En effet, au moment de la rédaction de ces lignes, nous estimons qu'il est le meilleur pour aborder les règles d'accessibilité, et ce, pour plusieurs raisons.

Tout d'abord, le référentiel AccessiWeb est le seul qui permette, lorsqu'on l'utilise, de valider à la fois les critères AccessiWeb, mais également les règles de succès WCAG 2 et les tests RGAA. On fait ainsi d'une pierre trois coups !

Par ailleurs, AccessiWeb dispose d'une communauté et bénéficie du retour d'expérience de nombreux utilisateurs depuis 2003 ainsi que de mises à jour quand cela s'avère nécessaire, contrairement au RGAA.

Sa rédaction sous forme de questions et son découpage en treize thématiques opérationnelles en font également un document plus facile à comprendre et à utiliser, de notre point de vue, que les WCAG 2 ou le RGAA.

La dernière raison qui nous conduit à préférer AccessiWeb, c'est sa licence. Alors que le RGAA est sous licence non commerciale (voir le chapitre 3), le référentiel AccessiWeb est libre d'utilisation¹.

À RETENIR Oui = validé

Un travail important a été réalisé pour rendre le référentiel AccessiWeb utilisable de façon simple et pratique. L'un des aspects les plus frappants réside dans la formulation sous forme de questions des critères AccessiWeb. À chaque question, si la réponse est oui, le critère est validé. Si la réponse est non, le critère est invalidé.

Cela peut paraître trivial, mais c'est pourtant l'une des caractéristiques majeures qui font du référentiel AccessiWeb le plus pédagogique et le plus simple à utiliser de notre point de vue.

Notons qu'il s'agit d'une particularité propre à AccessiWeb, qu'on ne retrouve ni dans les WCAG 2 ni dans le RGAA.

C'est pour toutes ces raisons que nous avons décidé de vous présenter en détail les règles d'accessibilité à travers le référentiel AccessiWeb. Mais sachez que vous pouvez très bien lire les chapitres qui suivent tout en décidant dans votre pratique quotidienne d'utiliser plutôt le RGAA ou directement les WCAG 2, selon votre préférence. L'important ici est de faire de l'accessibilité web, et nous allons vous aider à mieux comprendre les règles d'accessibilité.

1. Voir la section « Conditions d'utilisation » à l'adresse :
http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#intro

EN PRATIQUE Évaluer la conformité aux critères AccessiWeb 2.2

Dans les chapitres qui suivent, nous allons étudier dans le détail les concepts utiles pour bien comprendre chaque thématique AccessiWeb. Il ne s'agit donc pas de reprendre les éléments déjà fournis par AccessiWeb, disponibles en ligne.

Ainsi, si vous devez évaluer une page web, nous vous conseillons de lire d'abord ces chapitres, puis de suivre pas à pas le référentiel AccessiWeb 2.2 dans sa version déployée (avec les tests).

Vous pourrez utiliser une grille d'évaluation et déterminer, critère après critère, si dans le cas de la page web étudiée, le critère est validé, invalidé ou non applicable.

Pour la procédure de test, reportez-vous au document en annexe du référentiel intitulé « Méthodologies de tests AccessiWeb 2 ».

► <http://ur1.ca/9yks3^a>

a. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#annexe

Les chapitres 6 à 11 qui suivent présentent les treize thématiques AccessiWeb, regroupées plus ou moins arbitrairement selon les chapitres. Chaque section, consacrée à une thématique, comporte en introduction un rappel de la recommandation générale AccessiWeb avec sa référence web, ainsi que le bénéfice que représente cette thématique pour l'utilisateur final.

Vous trouverez par la suite le détail de ce qu'il faut retenir de la thématique (définitions, procédure d'évaluation, outils, cas d'exception, différences avec le RGAA le cas échéant), sans pour autant faire la liste des 133 critères qui composent le référentiel AccessiWeb. L'idée n'est pas de faire un copier-coller d'AccessiWeb (que nous vous invitons à lire par ailleurs), mais de vous donner les clés pour utiliser ce référentiel à bon escient.

ATTENTION Travailler sur le code source généré

Tout au long des chapitres 6 à 11, nous allons avoir recours au code source des pages web. Comme nous l'avons vu précédemment, les technologies d'assistance, en particulier les lecteurs d'écran, interprètent uniquement le code HTML. Mais attention, il s'agit du code HTML généré, c'est-à-dire après modification éventuelle du DOM en JavaScript.

Pour obtenir ce code, vous pouvez télécharger l'extension Web Developer pour Firefox, créée par Chris Pederick :

► <http://ur1.ca/9y9y9^a>

Une fois l'extension installée, une barre d'outils apparaît.

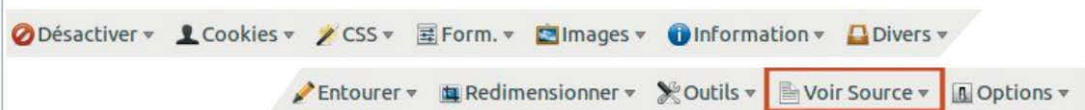


Figure 5-1 Barre d'outils de l'extension Web Developer

Dans l'onglet *Voir source*, sélectionnez *Voir le code source généré*.

a. <https://addons.mozilla.org/fr/firefox/addon/web-developer/>

Notre approche se fera donc thématique par thématique. Nous étudierons en détail certains critères quand cela s'avérera nécessaire, et passerons sous silence certains critères de niveau Or, par exemple, pour ne pas vous encombrer de détails qui ne vous seront sans doute pas utilisés dans votre pratique².

Enfin, en ce qui concerne les outils étudiés, le choix a été fait de vous présenter uniquement des outils dont les résultats sont fiables, que vous pouvez utiliser librement et que nous utilisons à titre personnel.

Il existe beaucoup d'autres solutions, libre à vous de les utiliser en vous assurant de leur fiabilité.

À SAVOIR **AccessiWeb 2.2 ne prend pas en compte HTML5**

La force, et la faiblesse, des WCAG 2 est qu'elles ne reposent sur aucune technologie. Les critères de succès sont donc applicables quelles que soient les techniques utilisées. Mais il est par conséquent très difficile de savoir comment faire concrètement, puisque la responsabilité de savoir ce qui est « compatible avec l'accessibilité » revient aux développeurs.

Le référentiel AccessiWeb 2.2 est beaucoup plus opérationnel, puisqu'il s'ancre dans la réalité des technologies utilisées. C'est la raison pour laquelle nous vous le présentons. Il présente toutefois une limite, à savoir qu'il repose exclusivement sur du HTML4 et du XHTML, ne prenant pas en compte les nouveautés du HTML5 qui fleurissent sur la Toile.

Tout ce que nous allons aborder dans ce chapitre concerne donc uniquement des encodages antérieurs à HTML5. Mais les grands principes énoncés restent valables quelle que soit la technologie utilisée.

Nous verrons dans le chapitre 12 ce qu'il en est de HTML5 et de l'accessibilité.

Notons qu'il est prévu à moyen terme que BrailleNet décline une nouvelle version d'AccessiWeb, dédiée à HTML5. Mais les travaux ne sont pas encore entamés. À suivre donc...

Les chapitres 6 à 11 qui suivent présentent donc successivement les différentes thématiques AccessiWeb :

- **Chapitre 6.** Thématiques 1 à 3 : Images, Cadres, Couleurs
- **Chapitre 7.** Thématique 4 : Multimédia
- **Chapitre 8.** Thématique 5 : Tableaux
- **Chapitre 9.** Thématiques 6 et 7 : Liens et Scripts
- **Chapitre 10.** Thématiques 8 à 10 : Éléments obligatoires, Structuration de l'information, Présentation de l'information
- **Chapitre 11.** Thématiques 11 à 13 : Formulaire, Navigation, Consultation

2. Les critères passés sous silence seront indiqués, libre à vous de les consulter par vous-même dans le référentiel.

Images, cadres et couleurs (thématiques AccessiWeb 1 à 3)

Il est temps de nous plonger dans les trois premières thématiques du référentiel AccessiWeb, à savoir les images (1), les cadres (2) et les couleurs (3).

Images

« Donner à chaque image porteuse d'information une alternative textuelle pertinente et une description détaillée si nécessaire. Remplacer les images textes par du texte stylé lorsque c'est possible. »

<http://ur1.ca/a0cbj>¹

Les images sont très fréquentes sur les pages web. Ce sont elles qui permettent de réaliser des mises en page avantageuses, elles véhiculent de l'émotion et parfois même des informations d'une façon que le texte seul ne peut transmettre.

Un article intéressant, s'il est mis en valeur par une image, sera d'autant plus agréable à lire. Or, les images font partie de ces rares éléments qu'on ne peut pas rendre accessibles nativement. Si une personne ne peut pas visualiser les images, la seule façon pour elle d'accéder à l'information sera d'en avoir une description, fournie par une alternative.

Évidemment, cette impossibilité de voir les images concerne en premier lieu les personnes aveugles. Mais plus largement, cette problématique touche également les personnes ayant une connexion bas débit rendant impossible l'affichage de certaines images trop lourdes, un smartphone ancienne génération n'affichant pas les images, ou encore certaines personnes travaillant sur des navigateurs texte.

1. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#images

Si un navigateur ne parvient pas à charger une image, ou si un logiciel de lecture d'écran détecte un élément image (balise `<img>`, par exemple), il essaiera de lire si une alternative a été renseignée. Il est ainsi possible de rendre accessible une image pour les personnes ne pouvant pas la voir.

Notons que les robots d'indexation des moteurs de recherche ne « voient » pas non plus le contenu des images et se servent donc des alternatives pour référencer un contenu image.

Les bénéficiaires d'une alternative aux images sont donc plus nombreux qu'il n'y paraît. Voyons en détail ce que dit AccessiWeb à ce sujet.

INFO Critères de la thématique « Images »

La thématique « Images » contient neuf critères, dont sept de niveau Bronze, un de niveau Argent et un de niveau Or.

Nous traiterons ici tous les critères présents dans la thématique.

Un attribut alt sur chaque image

Comme mentionné au chapitre précédent, les critères AccessiWeb se décomposent en critères de présence et critères de pertinence, afin de faciliter le travail d'évaluation en ne posant qu'une seule question à la fois.

Le premier critère concerne donc la présence technique d'un attribut `alt` sur toute balise pouvant véhiculer une image (balises `<img>`, `<area>`, `<input>` avec l'attribut `type="image"`, `<applet>`).

La nécessité d'avoir un attribut `alt` pour toutes les images, qu'elle que soit leur nature, est fondée sur le quatrième grand principe des WCAG : la robustesse.

Pour que le code soit interprété correctement tant par les navigateurs que par les technologies d'assistance, encore faut-il qu'il soit correctement construit. Et si l'attribut `alt` est une nécessité en termes d'accessibilité, il sert également à prévoir le comportement lorsque l'image ne peut pas être affichée.

Vient ensuite la question de ce qu'il faut indiquer dans cet attribut. Il s'agit alors de se poser la question de *la pertinence de l'alternative*. Et pour ce faire, encore faut-il savoir distinguer le type d'image auquel on a affaire.

OUTILS Détecter les éléments images sans attribut alt

Concrètement, il vous serait évidemment possible d'afficher le code source et d'utiliser le raccourci clavier **Ctrl+F** pour rechercher toutes les balises `<img />`, `<area>`, `<input>` avec l'attribut `type="image"` ou `<applet>`, puis de vérifier que chaque élément comporte bien un attribut `alt`. Il est heureusement possible d'éviter cet exercice fastidieux en ayant recours à des logiciels automatisant cette tâche. Nous en citerons deux.

- Le validateur du W3C, qui détecte toutes les erreurs de code. Si un des éléments mentionnés ne contient pas d'attribut `alt`, il sera détecté comme invalide.

► <http://validator.w3.org>

- Tanaguru, qui étudie chacun des tests du critère 1.1 et indique s'il est validé, non validé ou non applicable.

► <http://my.tanaguru.com>

Images porteuses d'information, images de décoration et images d'illustration

AccessiWeb distingue deux types d'images : les images porteuses d'information et les images de décoration.

Les images porteuses d'information sont les plus simples à distinguer : si elles ne sont pas présentes, on perd de l'information. Par exemple, si le titre du site contient une police de caractères spéciale, il est fréquent qu'il soit réalisé sous forme d'image. Si l'image ne s'affiche pas, le titre du site n'apparaît pas. Or, il s'agit d'un élément essentiel pour tout site web, porteur d'une information importante.

Les images de décoration, en revanche, sont des images servant à « faire joli » sur la page, sans autre but que de décorer. Les exemples donnés dans le glossaire d'AccessiWeb (<http://ur1.ca/9zrn42>) parlent d'eux-mêmes :

« Image n'ayant aucune fonction et ne véhiculant aucune information particulière par rapport au contenu auquel elle est associée. Exemples :

- une image servant à caler la mise en page ;
- une image de coin arrondie pour habiller un bloc d'information ;
- une image d'illustration n'apportant aucune information nécessaire à la compréhension du texte auquel elle est associée. »

AccessiWeb ne distingue donc que deux types d'images : les images de décoration et les images porteuses d'information. Ces deux types d'images sont clairement identifiables. Or, il existe un cas un peu trouble, dans lequel le doute s'insinue très vite, source de débats parfois insolubles : les images d'illustration.

2. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#mImgDeco

REMARQUE Images de décoration et CSS

Vous êtes peut-être en train de vous dire que les images de décoration doivent être placées dans les CSS... Vous avez raison et cela signifie que vous avez de très bonnes habitudes de codage, loin de nous l'idée de vous en éloigner. Appeler les images de décoration via l'utilisation de feuilles de styles est en effet une bonne pratique, que nous vous encourageons à mettre en œuvre. Toutefois, cela ne pose pas une question d'accessibilité, dans la mesure où cela ne gêne en rien l'accès à l'information. Car si l'attribut `alt` est vide, l'image sera ignorée, ce qui ne perturbera pas la lecture du lecteur d'écran. Si vous êtes dans l'obligation d'utiliser une image de décoration dans le code HTML plutôt que via des CSS, cela ne doit pas vous empêcher de rendre cette image accessible en ajoutant une alternative appropriée.

La définition du glossaire concernant les images de décoration mentionne les images d'illustration. Or, ces images sont en général ajoutées par les contributeurs via un logiciel de gestion de contenus (CMS) et sont souvent dans leur esprit porteuses d'information. Mais l'information restituée est la plupart du temps d'ordre émotionnel, plus qu'informatif.

Prenons un exemple concret. Sur le site du journal *Le Point*, un article publié le 11 mars 2011 (<http://ur1.ca/9y3uk>³) et intitulé « Tsunami dévastateur au Japon » relate le raz-de-marée qui a suivi le séisme dramatique qui ravagea le Japon (figure 6-1). Cette catastrophe fut l'objet de nombreuses photos et vidéos, permettant aux gens du monde entier de vivre ce drame. Une photo impressionnante introduit d'ailleurs l'article, précédée du texte suivant : « Un tremblement de terre record a provoqué sur les côtes nippones un raz-de-marée qui a tout emporté sur son passage. »

Un attribut `alt`, un attribut `title` (infobulle) ainsi qu'un texte de légende reprennent le texte suivant : « Une vague emportant tout sur son passage est remontée dans les terres. »

Or, cette alternative n'apporte rien par rapport au texte environnant. Et cette image d'illustration, qui véhicule pourtant un choc visuel d'une grande force, n'est là que pour illustrer le texte. Elle véhicule certes une émotion pour ceux qui peuvent la voir, mais aucune information supplémentaire n'est donnée par rapport au texte. Il est difficile de parler d'image de « décoration » et de placer ce genre de photo sur le même plan que les images servant à faire un coin arrondi. Pourtant, d'un point de vue purement informatif, elle n'apporte rien de plus que ce que l'article dit déjà dans son texte. Il convient donc de traiter cette image comme une image de décoration.

3. http://www.lepoint.fr/monde/tsunami-devastateur-au-japon-11-03-2011-1305103_24.php

Figure 6–1
Article publié sur le site
Le Point après le tsunami
au Japon en mars 2011

Le Point.fr

EN CONTINU
→ 24h d'info
→ Flux RSS
→ Mobile
→ Newsletters

LE MAGAZINE
→ Sommaire
→ Abonnement
→ Édition digitale
→ Nos hors-séries

LES SERVICES
→ Météo
→ Bourse
→ Jeux-Concours

ACTUALITÉ | *Débattre* | POLITIQUE | ÉCONOMIE | TECH & NET | SANTÉ | CULTURE | ART DE VI

Actualité | Politique | **Monde** | Société | Économie | Médias | Sciences | Villes | Sports | Insolite

ACTUALITÉ » Monde RSS

Le Point.fr - Publié le 11/03/2011 à 07:25 - Modifié le 11/03/2011 à 19:47

Tsunami dévastateur au Japon

Un tremblement de terre record a provoqué sur les côtes nippones un raz-de-marée qui a tout emporté sur son passage.



Une vague emportant tout sur son passage est remontée dans les terres. © Newscom / Sipa

À NOTER Les journaux et les images créditées

Dans les journaux, citer ses sources fait partie du métier. Et lorsqu'on est reporter-photographe, cela permet également de faire connaître son nom. Ceci explique pourquoi chaque image est toujours accompagnée d'une légende citant le nom du photographe. Or, cette information n'apporte rien en termes d'accessibilité. Surtout quand l'image est purement illustrative et ne fournit pas plus d'information que le texte qui l'accompagne. Cette information peut même nuire dans le sens où elle alourdit la lecture d'un article qui peut être complexe, texte que la « douce voix » de la synthèse vocale rend encore moins digeste... On voit là une opposition de points de vue concernant ces images d'illustration : d'un côté, les journalistes qui considèrent que chaque photo est une donnée dont il convient de mentionner la source, de l'autre côté, les lecteurs non voyants qui préféreraient que ces données soient ignorées de manière à alléger la lecture du texte de la page, ces images étant pour eux de la pure décoration...

Nous verrons que HTML5 répond en partie à ce problème, en introduisant une nouvelle balise `<figcaption>`.

Prenons un autre exemple, celui d'une image d'illustration publiée sur le site de Rue89 (figure 6-2). Dans l'article du 11 août 2012 (<http://ur1.ca/9xxqs>⁴) intitulé « Le plongeur, ce sport qui ne rend pas super beau », le texte rend compte de la tête que font les plongeurs en plein effort :

« Un plongeur, c'est tellement rapide qu'on n'a pas le temps de se concentrer sur le visage des sportifs. Heureusement, les appareils photos d'aujourd'hui capturent plusieurs images à la seconde.

Le *Guardian* s'est amusé à rassembler les plus belles têtes déformées de ces Jeux dans un diaporama bien rigolo. Et ça marche aussi en plongeur synchronisé, pour deux fois plus de plaisir. »

La photo d'un plongeur chinois faisant une grimace indescriptible ouvre cet article. La légende de cette image est la suivante : « Cao Yuan, un plongeur à 10 m, immortalisé en plein effort. (Torsten Blackwood/AFP) ».

L'image étant le sujet principal, suscitant le sourire des lecteurs, il est nécessaire d'en faire une vraie description. Le texte ne faisant que sous-entendre la tête que font les athlètes dans ces images, il est donc indispensable de décrire précisément ces photos. Ainsi, si un lecteur aveugle souhaite savoir pourquoi l'image est si drôle, il pourra en lire une description, humoristique de préférence pour rester dans le ton de l'article. Or, la légende telle qu'elle est décrite, n'est d'aucune utilité. Bien qu'il y ait une information associée à l'image, elle n'est pas pertinente. Dans ce cas, c'est moins l'image qui illustre l'article, que l'article qui accompagne l'image.

Notons que si le texte décrivait précisément la photo, l'image aurait été alors considérée comme non porteuse d'information par rapport au texte environnant, et aurait dû être traitée comme une image de décoration. Une même image d'illustration peut donc, selon le texte qui l'accompagne, être considérée comme porteuse d'information ou non. Tout dépend du contexte.

Nous avons vu à travers ces deux exemples, deux images d'illustration à traiter dans le premier cas comme une image de décoration, et dans le second cas comme une image porteuse d'information. Le contexte est essentiel pour qualifier une image d'illustration.

Retenons donc qu'il existe trois types d'images : les images porteuses d'information, les images de décoration et les images d'illustration, à traiter comme des images porteuses d'information ou comme des images de décoration selon le texte associé. Une fois le type d'image identifié, il convient de fournir une alternative pertinente.

4. <http://www.rue89.com/rue89-sport/2012/08/11/le-plongeur-ce-sport-qui-ne-rend-pas-super-beau-234542>

Figure 6–2

Article publié sur le site de Rue89 dont l'image représente un plongeur chinois en plein effort, faisant une grimace digne d'une compétition... comique !

EN IMAGES

11/08/2012 à 19h31

Le plongeur, ce sport qui ne rend pas super beau

Marine Marck | Rue89



<277

Aa - +



Cao Yuan, un plongeur à 10m, immortalisé en plein effort. (Torsten Blackwood/AFP)

Le plongeon est une discipline très spectaculaire : salto renversé groupé, double vrille, triple salto avant avec départ à l'équilibre... Les commentateurs énoncent les figures, on n'y comprend rien mais on observe avec admiration.

Ça nous paraît d'une grâce infinie. Le corps musclé, tendu, parfois imposant, se fait gracieux, léger et délié le temps d'un saut ([sauf quand ils se ratent](#)).

Un plongeur, c'est tellement rapide qu'on n'a pas le temps de se concentrer sur le visage des sportifs. Heureusement, les appareils photos d'aujourd'hui capturent plusieurs images à la seconde.

Le Guardian s'est amusé à rassembler les plus belles têtes déformées de ces Jeux dans un [diaporama bien rigolo](#). Et ça marche aussi en [plongeon synchronisé](#), pour deux fois plus de plaisir.

À NOTER Le cas des images-liens

Certaines images, en plus d'être présentes pour leur portée informative, descriptive ou illustrative, servent de porte d'entrée vers une autre page, un lien leur étant associé.

Dans ce cas, la fonction de l'image (le lien) l'emporte sur sa nature et le contenu de l'attribut `alt` doit être traité dans la thématique 6 consacrée aux liens. L'attribut `alt` d'une image-lien étant traité comme un intitulé de lien, la pertinence sera jugée à l'aune des critères 6.1 et 6.3 concernant les liens explicites.

Rédiger une alternative pertinente

À chaque type d'image, son alternative. Ce qui compte, c'est qu'elle soit pertinente, c'est-à-dire qu'elle doit fournir l'information utile à la compréhension de la page dans son contexte, et permettre aux technologies d'assistance d'ignorer l'image si celle-ci n'apporte pas d'information. *Il est aussi important d'éviter les informations inutiles, véritable pollution sonore lorsqu'on navigue à l'aide d'un lecteur d'écran couplé à une synthèse vocale, que de ne pas omettre des informations importantes.*

Une note dans le glossaire AccessiWeb indique par ailleurs qu'en présence d'un attribut `title`, et bien que cela ne relève pas du référentiel, il est conseillé que le contenu de l'attribut `title` soit identique au contenu de l'alternative.

ATTENTION Ces images porteuses d'information placées dans les CSS

Nous l'avons expliqué à plusieurs reprises, mais redisons-le ici : seul le code HTML généré par le navigateur est interprété par les technologies d'assistance. Ainsi, si vous insérez une image porteuse d'information dans une feuille de styles, vous ne pourrez pas ajouter d'alternative et l'information sera perdue. Cette problématique est traitée par le critère 10.2 de la thématique « Présentation de l'information » (voir le chapitre 10).

Description courte

La première des choses à faire pour ajouter une alternative à une image, consiste à renseigner son attribut `alt`. À la pertinence de l'alternative vient s'ajouter la nécessité de faire une description courte et concise.

À NOTER Longueur du texte des attributs alt

Nous avons indiqué que le texte des attributs `alt` se devait d'être court et concis. Cette nécessité est due au fonctionnement des technologies d'assistance, notamment les plages Braille et les logiciels d'agrandissement, qui rendent la navigation complexe sur de longues portions de texte.

AccessiWeb indique dans son glossaire une limite « recommandée » de 80 caractères. Cette limite est donnée à titre indicatif et peut être dépassée dans certains cas particuliers, par exemple lorsque l'image contient un nom propre très long qu'on ne peut réduire.

► <http://ur1.ca/9yktj^a>

a. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#mAltCC

Pour une image porteuse d'information, l'alternative doit décrire l'information véhiculée.

Pour une image de décoration, l'alternative doit être vide, ce qui se met en œuvre via un `alt=""` (notons l'absence d'espace entre les guillemets).

Pour une image d'illustration, tout dépend du contexte dans lequel elle se trouve :

Le contexte décrit la même information que celle véhiculée par l'image : l'image sera traitée comme une image de décoration avec un `alt` vide. Il s'agit de la grande majorité des cas.

Le contexte ne décrit pas la même information que l'image : l'image sera traitée comme une image porteuse d'information et son attribut `alt` devra être renseigné de façon appropriée.

CAS PRATIQUE Comment traiter les images ayant une légende ?

Lorsqu'une image est décrite par une légende, nous pouvons considérer que ce petit texte fait déjà office d'alternative. Ajouter un attribut `alt` reprenant le même contenu ne fera qu'alourdir la navigation.

Le problème de la légende, c'est que rien ne permet de la lier à l'image au niveau du code. Heureusement, cette problématique a été prise en compte et réglée en HTML5, ce que nous verrons au chapitre 12. En attendant, plusieurs écoles existent à ce sujet. Pour notre part, nous vous recommandons de vous concentrer sur la nature de l'image pour renseigner ou non l'attribut `alt`. Ainsi, si l'image n'apporte pas d'information, ajouter un attribut `alt` vide, même si elle est accompagnée d'une légende.

Quand l'image est un lien, l'attribut `alt` doit indiquer la destination de ce dernier ou sa fonction s'il permet d'effectuer une action.

Enfin, l'image peut également servir à faire barrage aux robots spammeurs, comme nous le verrons dans la section dédiée aux captchas ci-après.

OUTILS Afficher le texte des alternatives

La pertinence d'une alternative ne peut s'apprécier de façon automatisée et doit être évaluée par une personne compétente. Des outils permettent de procéder à cette évaluation en évitant de passer par un affichage du code source, qui rend la tâche fastidieuse.

Nous avons mentionné précédemment l'extension Web Developer pour Firefox. Dans la barre d'outils de cette dernière, sélectionnez ainsi *Afficher les attributs alt* dans le menu *Images*. Une étiquette noire sur fond jaune apparaît alors sur la page avec, en face de chaque image, le texte de l'attribut `alt` s'il est renseigné, ou uniquement la mention `alt` si l'attribut est vide.

Description longue

Dans certains cas, l'image véhicule une information bien trop complexe pour être restituée en 80 caractères. Une description longue est alors nécessaire. Elle ne sera

Figure 6-3

Page des WCAG 2 avec affichage des attributs alt au moyen de la barre d'outils de Web Developer : on voit que l'alternative au logo est renseignée avec la mention W3C.

W3C Recommendation



[\[accès aux contenus\]](#)

Web Content Accessibility Guidelines (WCAG) 2.0

pas fournie par l'attribut `alt`, mais de préférence grâce à un lien adjacent à l'image menant à une page web détaillant l'information véhiculée par l'image.

C'est le cas notamment lorsqu'une image telle qu'un graphique ou un plan est insérée dans un article. À la condition bien sûr que cette image d'illustration ne soit pas déjà explicitée par le texte situé autour.

Figure 6-4

Page d'accueil de la Bergerie de Villarceaux, avec un plan et sa description détaillée via un lien adjacent.

(<http://bergerie-villarceaux.org>)



Notons que la présence d'une description longue implique qu'il y ait également un attribut `alt` renseigné.

ATTENTION Attribut `longdesc` déprécié en HTML5

Rappelons que le référentiel AccessiWeb 2.2 est basé sur les versions antérieures à HTML5. Ainsi, bien qu'il autorise l'utilisation d'un attribut `longdesc`, qui permet d'indiquer l'adresse d'une page sur laquelle sera donnée la description détaillée de l'image, HTML5 interdit l'utilisation d'un tel attribut. Par ailleurs, les utilisateurs non voyants savent rarement comment afficher le contenu de cet attribut. Mieux vaut donc utiliser un autre moyen, tel qu'un lien adjacent à l'image. Dans la mesure où il n'est pas caché dans le code, il bénéficiera à tous.

Cas particuliers : captchas et images tests

Comme nous l'avons évoqué en première partie de ce livre : l'accessibilité ne vise pas à interdire quoi que ce soit, ni à brider la créativité, mais simplement à donner accès à tous aux mêmes contenus et fonctionnalités.

VOCABULAIRE Captcha

Ce mot barbare est en réalité l'acronyme de *Completely Automated Public Turing test to tell Computers and Humans Apart*, qui signifie en français « test de Turing complètement automatique pour différencier les humains et les robots ».

Décrit par Alan Turing en 1950, Wikipédia indique que « le test de Turing est une proposition de test d'intelligence artificielle fondée sur la faculté d'imiter la conversation humaine ».

Pour en savoir plus sur les captchas, consultez le site de Wikipédia à l'adresse suivante :

► <http://ur1.ca/9yuk7^a>

Si vous souhaitez davantage d'informations sur le test de Turing, visitez cette page Wikipédia :

► <http://ur1.ca/39owu^b>

Enfin, n'hésitez pas à consulter le site officiel consacré aux captchas (en anglais) :

► <http://captcha.net>

a. <http://fr.wikipedia.org/wiki/CAPTCHA>

b. http://fr.wikipedia.org/wiki/Test_de_Turing

Les captchas consistent à mettre en place un système permettant de lutter contre l'envoi abusif d'e-mails par des robots, en utilisant des images avec un code (souvent des lettres et des chiffres déformés).

Figure 6-5

Exemple de captcha sur le site de démonstration reCAPTCHA (<http://www.google.com/recaptcha/learnmore>)

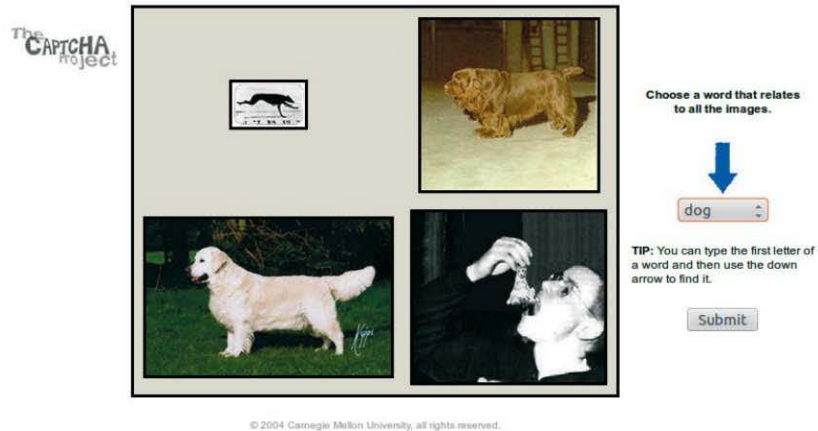


L'objectif est de vérifier que la personne n'est pas un robot mais bien un être humain. Il est alors demandé à l'internaute de recopier ce code, souvent illisible. Le but de ce mécanisme est de contrer les robots ne sachant pas lire les images, et de contrer les processus de reconnaissance optique des caractères (OCR), d'où la déformation des images.

Il peut aussi être demandé de reconnaître une image, ou de choisir dans une liste déroulante un qualificatif pour décrire ce que des images présentées ont en commun, comme dans l'exemple de la figure 6-1 (l'internaute doit indiquer dans la liste que le point commun de ces images est qu'elles représentent toutes un chien).

Figure 6-6

Exemple de captcha reposant sur la reconnaissance d'images, sur le site de ESP-PIX (<http://ur1.ca/9yuli>)



© 2004 Carnegie Mellon University, all rights reserved.

a. <http://server251.theory.cs.cmu.edu/cgi-bin/esp-pix/esp-pix>

Or, si un attribut `alt` donne la réponse en indiquant le contenu de l'image, les robots seront capables de le lire et l'effet recherché sera anéanti.

L'accessibilité ne visant pas à interdire une pratique, mais seulement à donner accès à tous, sans discrimination, aux mêmes contenus et fonctionnalités, il est demandé deux choses :

- indiquer, via l'attribut `alt`, la nature et la fonction de l'image ; dans le cas d'un captcha, l'alternative pourrait donc être, selon le type de captcha, `alt="Code à recopier pour éviter les robots spammeurs"` ;
- proposer un autre moyen, accessible, permettant d'accéder au contenu. Il existe différents moyens alternatifs : des captchas audio, à peu près aussi inaudibles que les captchas images sont illisibles, ou encore des captchas sous forme de questions auxquelles il faut donner une réponse (résoudre un calcul simple, par exemple, mais se pose alors le problème du handicap cognitif comme la dyscalculie). Tout moyen est valide, comme la solution proposée par Wikipédia fournissant la possibilité de s'inscrire par e-mail en cas d'impossibilité d'utiliser le captcha (figure 6-7). D'autres méthodes existent, comme l'envoi du code par SMS – libre à chacun d'effectuer ses recherches selon ses besoins.

Créer un compte ou se connecter

Créer un compte

Vous avez déjà un compte ? [Connectez-vous.](#)

Comme protection contre les créations de compte abusives, veuillez entrer les mots ci-dessous dans la boîte ([plus d'informations](#)). Si l'image ne s'affiche pas ou que vous avez un problème, vous pouvez envoyer un courriel à info-fr@wikimedia.org. N'oubliez pas de spécifier le nom d'utilisateur que vous désirez pour la création de compte assistée



CAPTCHA :

Figure 6-7 Exemple de captcha pour créer un compte sur Wikipédia : une alternative est proposée avec la possibilité de s'inscrire par e-mail si la lecture du code est impossible.

Prenons un exemple concret et qui ne manque pas d'ironie. Le site ministériel dédié à l'accessibilité et à la conception universelle propose de trouver les coordonnées du correspondant accessibilité de son département. Si l'on clique pour obtenir les coordonnées du correspondant dans le Val d'Oise, on arrive sur une page représentant un exemple typique de ce qu'il ne faut pas faire (figure 6-8).

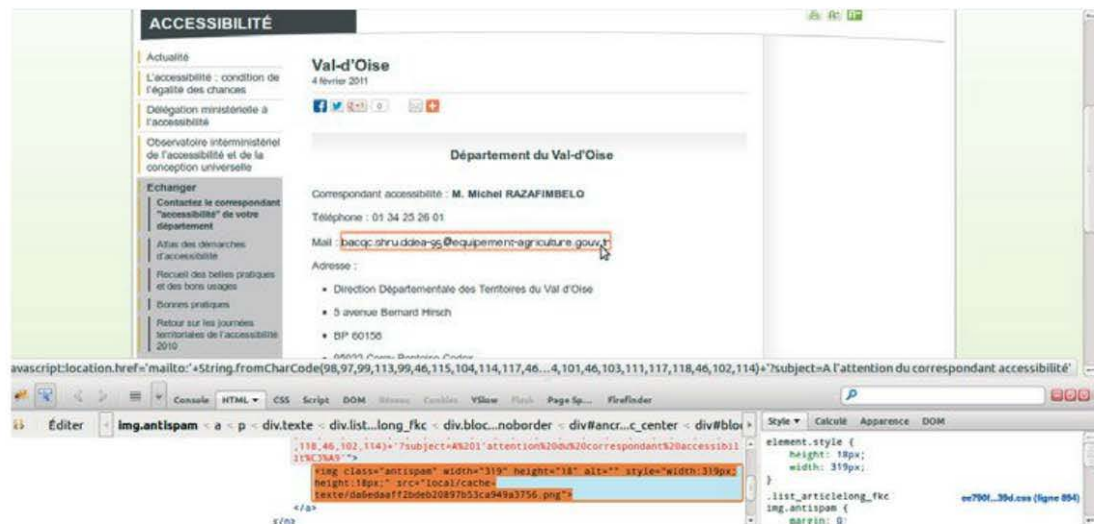


Figure 6-8 La page de contact du correspondant accessibilité dans le Val d'Oise : l'adresse-mail est une image avec un attribut alt vide. (<http://ur1.ca/acgxlk>)

a. <http://www.developpement-durable.gouv.fr/Val-d-Oise,20104.html>

Pour éviter les spams, l'adresse électronique du correspondant est insérée dans la page sous forme d'image avec un `alt` vide. Dans la mesure où il n'existe aucun autre moyen d'obtenir l'adresse du correspondant, cette information est inaccessible à quiconque ne peut voir les images. Ce qui est un comble pour un service dédié à l'accessibilité et la conception universelle ! Le dispositif est donc efficace pour dissuader

toute personne aveugle de contacter son correspondant départemental. Il est malheureusement peu probable que cela suffise à dissuader les spams...

DÉBAT Captcha : protéger les contenus des robots... ou des internautes ?

Ne vous est-il jamais arrivé de vouloir entrer dans un site sans y parvenir ? Vous vous retrouvez devant une image avec un code déformé tellement illisible que vous n'arrivez pas à le déchiffrer. Vous faites une première tentative. Échec. Une deuxième. Deuxième échec. Et là, vous abandonnez, frustré, pestant contre ce maudit site qui ne veut pas de vous...

Quand on voit l'ingéniosité de certains captchas, tous plus compliqués les uns que les autres, on peut parfois se demander de qui les éditeurs de sites souhaitent se prémunir... Des moteurs ou de leurs utilisateurs ?

Outre les questions d'accessibilité parfois épineuses que peuvent poser ces dispositifs, ils représentent souvent un véritable frein pour les utilisateurs, indépendamment d'une situation de handicap.

Par ailleurs, leur efficacité est de plus en plus souvent mise en cause. S'il s'agit avant tout de contrer des robots indexant automatiquement certaines adresses e-mails, on sait aujourd'hui que des sociétés ont maintenant recours à de véritables personnes pour entrer manuellement le code des captchas.

Nous ne nions pas le fléau que représente le spam sur Internet, et il est normal de vouloir s'en prémunir. Mais avant de déployer tout un arsenal antirobots, il serait judicieux de s'interroger sur leur pertinence et le coût que cela fait peser en termes de pertes d'utilisateurs...

Dans tous les cas, pensez à toujours prévoir une alternative accessible à tous.

Grâce à cet exemple, nous voyons comment la mise en accessibilité opère de manière générale. Il ne s'agit jamais de demander une solution identique dans tous les cas. La logique de mise en accessibilité doit s'adapter aux techniques mises en place. Ainsi, si le rôle d'une image est de ne pas être lue par un robot, cela n'aurait pas de sens de demander d'ajouter une alternative. C'est le cas pour les captchas, mais cela pourrait également être le cas pour un quiz dans lequel il faut trouver la bonne image. Voici ce qu'en disent les WCAG :

« Si le contenu non textuel est un test ou un exercice qui serait invalide s'il était présenté en texte, alors l'équivalent textuel fournit au moins une identification descriptive du contenu non textuel. »

Donner l'alternative correspondant à la réponse ruinerait l'expérience du jeu pour les utilisateurs aveugles.

La seule préoccupation se résume donc à fournir les mêmes contenus et les mêmes fonctionnalités sans discrimination. Nous verrons que la gestion des cas particuliers permet de gérer les critères d'accessibilité de façon suffisamment souple pour que l'accessibilité ne soit pas perçue comme un frein, mais au contraire comme une opportunité pour améliorer ses pages.

Les textes sous forme d'image

Nous abordons ce point rapidement, les deux critères (Argent et Or) sur lesquels il repose étant moins du ressort de l'accès à l'information que de celui d'un confort d'utilisation.

Les deux critères 1.8 et 1.9 avancent à peu près la même chose : les textes sous forme d'image doivent être remplacés par du texte stylé en CSS. Ce qui change, c'est le niveau d'exigence. Pour le critère 1.8, de niveau Argent, il est demandé de remplacer l'image texte par du texte stylé, *si possible* et *seulement si* un mécanisme de remplacement (telle une conversion des images en texte via JavaScript) n'est pas déjà en place. Le critère 1.9, de niveau Or, reprend la même exigence, si possible, mais un mécanisme de remplacement ne suffit pas à valider ce critère.

Nous retrouverons souvent cette dichotomie entre des critères ayant quasi le même intitulé, mais avec un niveau d'exigence différent.

Pour l'utilisateur, l'objectif de ces critères est double.

- **Permettre un agrandissement via les logiciels de zoom.** Les logiciels ont évolué et permettent aujourd'hui l'agrandissement des images comme des textes. Cependant, une image publiée sur le Web est souvent au format bitmap, c'est-à-dire composée de petits carrés qui deviennent visibles si l'image est trop agrandie, ce qui la rend floue quand elle est lissée. L'image est alors pixellisée. L'utilisation d'un texte stylé évite cet écueil et permet aux personnes malvoyantes d'agrandir le texte sans qu'il ne devienne flou.
- **Éviter de surcharger la lecture via un lecteur d'écran avec des informations inutiles.** En effet, comme nous l'avons vu dans le chapitre 2, les lecteurs d'écran décrivent à l'utilisateur non seulement ce qui est écrit, mais également la nature des éléments : images, liens, listes à puces... Si un texte est réalisé sous forme d'image (par exemple de cette façon : ``), la synthèse vocale couplée au logiciel de lecture d'écran lira « image : Titre de l'article ». La lecture d'un texte stylé est donc plus fluide, puisque la mention image n'est plus indiquée.

DÉBAT Pourquoi « si possible » ? Avec CSS3, presque tout est possible, non ?

Le cas le plus répandu d'image texte, autorisé car aucune alternative n'était possible jusque-là, concerne les titres ayant une police de caractères particulière, différente des polices système.

En effet, il est possible d'indiquer dans la feuille de styles le nom d'une police de caractères à utiliser sur la page, et c'est le navigateur du visiteur qui est chargé d'aller fouiller dans son ordinateur pour récupérer la police mentionnée. S'il ne la trouve pas, il en choisit une autre. Ainsi, si le graphiste a prévu une jolie police de caractères originale, mais qu'elle n'est pas installée sur l'ordinateur de l'internaute, toute la mise en page risque d'être altérée.

Le seul moyen pour s'assurer que tout le monde puisse lire la page web sans que la mise en page ne soit modifiée consistait donc à utiliser une palette restreinte de polices de caractères, présentes par défaut sur la plupart des systèmes d'exploitation. Les développeurs utilisaient donc les polices système et quand ils souhaitaient davantage d'originalité, ils utilisaient alors une image texte.

Or, aujourd'hui, la propriété CSS `@font-face`, déjà présente en CSS 2.1 mais qui n'était pas implémentée partout, permet de charger une police sur le serveur web et de demander au navigateur d'utiliser cette police en ligne. Du coup, le problème ne se pose plus de savoir si l'internaute a bien la police sur son ordinateur.

Par ailleurs, avec les propriétés CSS3 telles que `transform` ou `text-shadow`, les possibilités sont infinies et le recours aux images pour obtenir un effet particulier est de moins en moins nécessaire.

On peut donc se demander dans quelle mesure les critères 1.8 et 1.9 peuvent aujourd'hui être considérés comme inapplicables... Si tout est possible, ou presque, le critère 1.8 devrait toujours être applicable et il ne devrait plus y avoir que du texte stylé, non ?

Heureusement, l'interprétation peut être plus souple. Il y a bien des cas, pour des raisons de droit d'auteur – la police ne peut pas être mise à disposition sur un serveur – ou de performance, par exemple, où les images textes sont nécessaires, même s'il est techniquement possible d'insérer le texte sous forme de texte stylé sans recourir à une image. C'est notamment le cas quand charger toute une police de caractères pour styler un seul titre pèse plus lourd que d'utiliser une image texte. Notons que cette hypothèse relative à la performance n'est pas considérée par AccessiWeb comme un cas particulier permettant de s'affranchir de l'obligation posée par le critère 1.8. Mais, si vous n'êtes pas dans une optique de labellisation, c'est un critère que vous pouvez prendre en compte sans réellement altérer l'accès à l'information. Comme nous l'avons maintes fois répété, l'accessibilité n'est pas un frein, mais une opportunité pour améliorer ses pages et permettre à tous d'y accéder.

Une fois que vous aurez compris comment fonctionnent les règles d'accessibilité et à quoi elles servent, à vous de décider, en conscience, comment les prioriser par rapports aux autres contraintes de votre projet.

Cas particuliers

Parfois, même s'il est possible de styler le texte en CSS, il est permis de recourir aux images. C'est le cas quand le texte fait partie d'un logo ou d'un élément associé à l'identité graphique d'un organisme ou d'une société (un slogan, par exemple). Dans ce cas, il ne sera pas demandé de reproduire le logo sous forme de texte. En tous cas au niveau Argent (critère 1.8). En revanche, le critère 1.9 de niveau Or ne souffre aucune exception.. Vous comprenez sans doute mieux maintenant pourquoi le niveau Or n'est pas recommandé...

Un cas particulier a également été créé pour les polices sous droit d'auteur. Nous aborderons ce point plus en détail dans la thématique 4 « Multimédia » (chapitre 7).

Cadres

« Donner à chaque cadre et chaque cadre en ligne un titre pertinent. »

<http://ur1.ca/a0cb5>

Si l'on peut se réjouir de la disparition progressive des cadres (balise `<frame>`) en matière de mise en page, les cadres en ligne (`<i>frame</i>`) sont en revanche très utilisés pour insérer un contenu tiers dans une page web.

Lorsqu'on insère un cadre ou un cadre en ligne, on permet à l'utilisateur d'avoir sur cette même page le contenu d'une autre page. Or, pour qu'un utilisateur non voyant puisse choisir d'entrer ou non dans ce cadre, ce qui nécessite une manipulation de sa part, encore faut-il qu'il puisse savoir ce qu'il contient. C'est tout l'objet des critères AccessiWeb de cette thématique.

INFO Critères de la thématique « Cadres »

La thématique « Cadres » ne contient que deux critères, tous deux de niveau Bronze.
Nous traiterons ici tous les critères présents dans la thématique.

Un attribut title pertinent

Le critère 2.1 est un critère de présence (« Chaque cadre et chaque cadre en ligne a-t-il un titre de cadre ? ») et le critère 2.2 un critère de pertinence (« Pour chaque cadre et chaque cadre en ligne ayant un titre de cadre, ce titre de cadre est-il pertinent ? »).

Renseigner de façon pertinente l'attribut `title` revient à décrire le contenu du cadre afin d'indiquer sa nature, orienter l'utilisateur et lui permettre de cliquer ou non à l'intérieur. Rien de bien sorcier en somme.

5. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#cadres

Profitions de cette thématique pour noter que le référentiel AccessiWeb francise tout, ce qui peut dérouter les développeurs au premier abord. Heureusement, tous les termes sont accompagnés d'un lien vers un glossaire dans lequel ils sont explicités.

Un cadre correspond à la balise `<frame>`, un cadre en ligne à la balise `<iframe>` et un titre de cadre à l'attribut `title`.

Prenons un exemple : le site de la ville de Bordeaux contient une balise `<iframe>` permettant d'afficher les dernières vidéos publiées sur sa page Dailymotion (figure 6-9).

Figure 6-9

Site de la ville de Bordeaux
contenant un encart vidéo tiré
de Dailymotion
(<http://ur1.ca/ach08a>)



a. <http://www.bordeaux.fr/p63898/videos>

Le code de la balise `<iframe>` est le suivant :

```
<iframe marginheight="0" src="http://www.dailymotion.com/videozap/user/
villedebordeaux/1?position=bottom&cols=4&rows=1"
marginwidth="0" frameborder="0" height="381" scrolling="no"
width="376"></iframe>
```

Pour valider les critères relatifs à la thématique « Cadres »⁶, il aurait suffi d'ajouter un attribut `title` comme suit :

```
<iframe title="vidéos de la ville de Bordeaux sur Dailymotion"
marginheight="0" src="http://www.dailymotion.com/videozap/user/
villedebordeaux/1?position=bottom&cols=4&rows=1"
marginwidth="0" frameborder="0" height="381" scrolling="no"
width="376"></iframe>
```

6. Notons que même en ajoutant l'attribut `title`, le code n'est pas valide car il utilise des attributs de mise en page... Voir à ce sujet la section « Présentation de l'information » du chapitre 10, page 207.

Copier-coller = danger !

La recommandation a beau être très simple, et absolument nécessaire en termes d'accessibilité, elle est rarement suivie. Pourquoi ? Parce que le code inséré est souvent copié-collé depuis les outils proposés par les réseaux sociaux tels que Facebook, Dailymotion, Twitter, etc., et que les codes proposés ne sont jamais valides.

La difficulté réside donc moins dans la contrainte technique que dans l'organisation interne et la formation des contributeurs.

Voici deux pistes à explorer, outre le fait de former ses contributeurs bien sûr (voir le chapitre 13) :

- écrire aux grands diffuseurs de services de partage de contenus que vous utilisez pour les sensibiliser et leur demander de corriger leur code ;
- mentionner dans votre déclaration de conformité ou votre page accessibilité que les services fournis par tel éditeur sont inaccessibles. Une mauvaise presse peut être dissuasive...

À vous de jouer !

Couleurs

« Ne pas donner l'information uniquement par la couleur et utiliser des contrastes de couleurs suffisamment élevés. »

<http://ur1.ca/achb5>⁷

La couleur est directement liée au principe WCAG de la perception. La perception d'une page web sur un écran dépend de plusieurs facteurs : le matériel utilisé, le calibrage de l'écran, la luminosité... Et bien sûr, la perception des couleurs par chacun.

CULTURE La perception des couleurs est culturelle !

Une expérience passionnante a fait l'objet d'un documentaire (en anglais) diffusé sur la chaîne BBC. On y découvre qu'avec le langage, nous façonnons une perception des couleurs qui influence notre vision du monde.

La tribu Himba, au nord de la Namibie, est ainsi prise en guise d'exemple. Les Himbas utilisent des mots différents de ceux utilisés par les Occidentaux pour décrire les couleurs, mais voient-ils la même chose ? Au cours d'une expérience, un chercheur montre une roue de couleurs composée de douze carrés, où seul l'un des carrés est d'une teinte différente.

7. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#couleurs

Quand le chercheur montre une palette avec des carrés verts, qui semblent identiques, les Himbas n'ont aucune difficulté à trouver le carré ayant une nuance différente. Ceci est dû au fait qu'ils utilisent des mots différents pour décrire les deux couleurs vertes en présence. Si pour nous, Occidentaux, il est presque impossible de voir la différence, elle est limpide pour les Himbas.

Par contre, lorsque le chercheur montre un autre écran avec des carrés verts et un bleu, les Himbas n'arrivent pas à distinguer le carré de couleur différente. Ils utilisent en effet le même mot pour décrire ce vert et ce bleu, alors qu'il nous saute aux yeux, à nous, Occidentaux, que le carré bleu est d'une couleur différente des carrés verts.

En résumé, la perception des couleurs est liée au langage et à notre culture. On voit qu'il y a un rapport très personnel à la perception des couleurs, et à notre perception du monde en général, en dehors même d'un handicap.

Découvrez ce reportage passionnant de moins de huit minutes à l'adresse suivante :

► <http://ur1.ca/9y527>^a

a. http://www.youtube.com/watch?feature=player_embedded&v=4b71rT9fU-I

Le facteur humain est donc primordial. Certaines personnes ne voient pas certaines couleurs, d'autres voient tout en noir et blanc, d'autres encore voient des couleurs différentes que celles affichées ou n'arrivent pas à lire un contenu insuffisamment contrasté.

La question des couleurs est essentielle en termes d'accessibilité pour permettre à chacun d'accéder aux contenus et fonctionnalités des pages web.

INFO Les critères de la thématique « Couleurs »

La thématique « Couleurs » contient quatre critères, dont deux critères de niveau Bronze, un critère de niveau Argent et un critère de niveau Or.

Nous traiterons ici tous les critères présents dans la thématique.

Alternative aux informations données par la couleur

La couleur est très utile pour transmettre certaines informations à l'utilisateur. Par exemple, le rouge est souvent associé à un danger ou à une erreur.

L'accessibilité n'interdit pas d'utiliser les couleurs au motif que certaines personnes ne peuvent pas les percevoir. Il s'agit ici de s'assurer que l'information donnée par la couleur est également perceptible quand on ne la voit pas. C'est ce que s'attachent à vérifier les deux premiers critères de cette thématique (3.1 et 3.2), relatifs à la présence et à la pertinence d'une alternative pour les informations véhiculées par la couleur.

Les exemples foisonnent sur le Web, celui que nous vous présentons ci-après est très représentatif et très fréquent (figure 6-10). En effet, les formulaires comportent très souvent des champs obligatoires, ces derniers pouvant être indiqués par une couleur différente.

Figure 6–10
À la fin de ce formulaire d’inscription, on trouve la mention suivante : « en rouge les champs obligatoires ». (http://ur1.ca/ach1y^a))

a. http://www.salariance.com/full_inscript.php

Le présent ouvrage étant édité dans une collection en noir et blanc, vous êtes en ce moment même en situation de handicap puisque vous ne pouvez pas discerner les couleurs de la figure précédente... Impossible donc de savoir quels sont les champs à renseigner obligatoirement !

Une solution simple aurait été d’ajouter un signe distinctif comme un astérisque en plus de la couleur rouge. Ceci s’avère être une bonne pratique, illustrée par la figure 6-11.

Figure 6–11
Page d’inscription au forum du site Alsacrations, où il est indiqué : « Les champs obligatoires sont identifiés par un astérisque (*) ». Ainsi, l’information importante n’est pas véhiculée que par la couleur. (http://ur1.ca/ach2r^a))

a. <http://forum.alsacreations.com/user.php>

À SUIVRE En savoir plus sur les formulaires

La thématique 11 du référentiel AccessiWeb est consacrée à l'accessibilité des formulaires. Pour en savoir plus sur ce point, reportez-vous au chapitre 11.

Un autre exemple d'inaccessibilité liée à l'emploi des couleurs, aussi fréquemment rencontré sur le Web, consiste à indiquer par la couleur l'étape en cours lors d'un processus d'achat en ligne ou à signaler des articles en promotion à l'aide d'une couleur, etc. C'est le cas par exemple sur le site d'Air France, où lorsqu'on achète un billet, les différentes étapes du processus d'achat sont signalées par une couleur différente, sans qu'une alternative ne soit disponible (figure 6-12).

Figure 6-12
Réservation sur le site d'Air France : on voit ici que nous sommes dans l'étape de recherche.
(<http://ur1.ca/ach3z>)



a. <http://www.airfrance.fr/cgi-bin/AF/FR/fr/local/process/standardbooking/BookNewTripAction.do>

Notons que la couleur peut être indifféremment portée par du texte (un mot ou un groupe de mots, par exemple), une image placée dans le code HTML, une image insérée au moyen des CSS, un objet multimédia telle qu'une animation... Il ne s'agit pas ici de détecter un élément HTML en particulier, mais bien d'évaluer la portée des couleurs dans la page.

Une fois identifié qu'une information est véhiculée par une couleur, il convient de déterminer quelle est précisément cette information et s'il existe dans la page un autre moyen, accessible, d'en prendre connaissance.

À ce stade, deux possibilités s'offrent à vous :

- absence d'alternative : le critère 3.1 est invalidé et le critère 3.2 sur la pertinence de l'alternative est alors non applicable. On ne saurait en effet invalider un critère sur la pertinence d'une alternative, si cette dernière n'existe pas ;
- présence d'une alternative : le critère 3.1 est validé. Il convient alors de vérifier si cette alternative est accessible afin de valider le critère 3.2. Dans le cas contraire, le critère 3.2 est invalidé. Voyons cela plus en détail.

OUTIL Visualiser une page web sans les couleurs

Encore une fois, seul un examen manuel peut permettre de détecter si des informations dans la page sont véhiculées par la couleur. Cependant, certains outils permettent de visualiser la page en noir et blanc ou selon différents troubles de la vue. À noter que visualiser la page en noir et blanc la fait apparaître en niveaux de gris. Les nuances de couleurs restent ainsi perceptibles, ce qui n'apporte rien dans le cadre d'un test portant sur l'information véhiculée par la couleur.

Il existe différents outils en ligne, ainsi que des extensions, mais ils ne sont pas toujours au point. Par exemple, l'outil Colorblind Web Page Filter permet de simuler l'absence de couleur :

► <http://ur1.ca/9yaon>^a

Vous pouvez également désactiver les couleurs via le menu *Désactiver* de la barre d'outils Web Developer, mais le résultat obtenu ne permet pas d'évaluer de façon pertinente la portée d'une information véhiculée par la couleur.

À titre personnel, dans notre pratique quotidienne, nous nous contentons d'un examen de la page sans outil autre que nos yeux et notre tête. En règle générale, méfiez-vous des outils permettant d'effectuer ce genre de test automatiquement. Vous seul pouvez juger si les outils existants vous sont d'une réelle utilité en fonction de vos besoins.

a. <http://colorfilter.wickline.org/>

Pertinence de l'alternative

Le glossaire AccessiWeb (<http://ur1.ca/9ykud>⁸) donne le contre-exemple suivant concernant la pertinence d'une alternative à une information donnée par la couleur :

« Dans le cas d'une liste d'articles dont les articles en jaune sont en promotion, l'utilisation de texte caché via CSS est un moyen de récupérer l'information " en promotion ", mais il n'est pas pertinent car cette information restera cachée à l'utilisateur qui visualise la page CSS activée. »

Cet exemple est intéressant, car il rappelle bien que la thématique des couleurs ne concerne pas que les personnes non voyantes, mais également les personnes jouissant d'une bonne vue mais ayant une incapacité à voir certaines couleurs. Si on utilise un texte caché via CSS comme alternative, les personnes utilisant un lecteur d'écran pourront accéder à cette alternative (à condition que le texte ait été caché de façon accessible...), mais celles lisant la page sans utiliser de technologie d'assistance n'y auront pas accès. L'alternative n'est pas accessible à tout le monde, elle n'est donc pas pertinente...

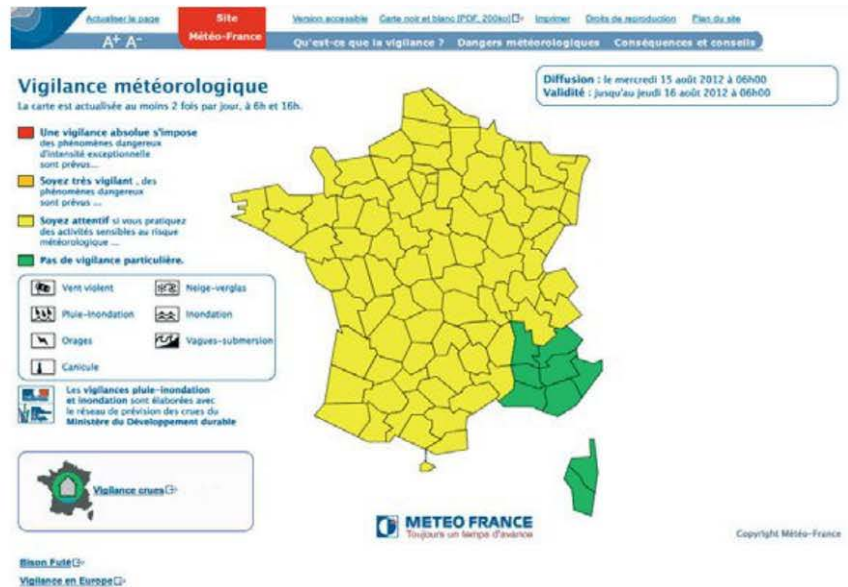
Voici un exemple d'une carte de Météo France montrant les zones de vigilance selon un code couleur allant du vert (pas de vigilance particulière) au rouge (une vigilance absolue s'impose) (figure 6-13, page suivante).

L'information est très clairement véhiculée par la couleur, sans aucun doute possible. Mais sur la figure, en noir et blanc, il est impossible de savoir quelles sont les zones à risque.

8. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#mPertinence

Figure 6-13

Carte de vigilance de Météo France : en vert, les zones sans risque ; en jaune, les zones présentant un risque modéré. (<http://ur1.ca/6pveg>)



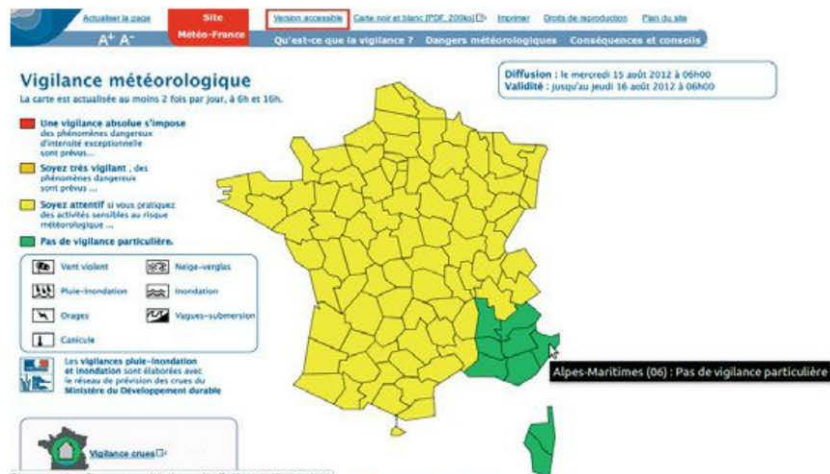
a. <http://france.meteofrance.com/vigilance/Accueil>

Le critère 3.1 est donc applicable. Mais y a-t-il une alternative à la couleur ?

À première vue, non. Or, en passant la souris sur la carte, une infobulle apparaît, grâce à l'utilisation de l'attribut `title` sur les zones réactives de l'image. Et en étudiant le code source pour vérifier si cette information est également accessible aux personnes non voyantes, on constate que l'information est également présente au sein des attributs `alt`. Cependant, pour que cela suffise à servir d'alternative, il aurait fallu que l'ensemble de la page soit accessible, ce qui n'est pas le cas. Conscients de cela, les responsables du site de Météo France mettent à disposition une version accessible via un lien sur la page (figure 6-14).

Figure 6-14

Au survol, la carte laisse apparaître une infobulle. Mais ceci étant insuffisant, une version accessible de la carte est disponible grâce à un lien en haut de la page.



Cette version accessible (figure 6-15) dresse la liste des départements, répartis selon les critères de vigilance. Une définition de chaque type de vigilance est également disponible. Ainsi peut-on voir sur la figure suivante que 88 départements sont en vigilance jaune, ainsi que la définition de ce niveau de vigilance. Puis s'ensuit la liste des départements, chacun étant accompagné d'un lien permettant d'accéder à plus de détails.

Vigilance météorologique

Diffusion : mercredi 15 août 2012 à 06h00
Validité : jusqu'au jeudi 16 août 2012 à 06h00
 Les informations de vigilance sont actualisées au moins 2 fois par jour, 6h et 16h

Accès rapide à un département
 Choisir un département

88 départements en vigilance jaune

Masquer la définition de la vigilance jaune

Vigilance jaune : Soyez attentifs; si vous pratiquez des activités sensibles au risque météorologique à proximité d'un rivage ou d'un cours d'eau; des phénomènes habituels dans la région mais occasionnellement et localement dangereux (ex. mistral, orage d'été, montée des eaux, fortes vagues submergeant le littoral) sont en effet prévus; tenez-vous au courant de l'évolution de la situation.

[Ain](#)
[Aisne](#)
[Allier](#)
[Ardèche](#)
[Ardennes](#)
[Ariège](#)

Figure 6-15 Version accessible de la carte de vigilance sur le site de Météo France : l'information véhiculée par la couleur est ici reproduite sous une forme textuelle.

En résumé, il est donc nécessaire de s'assurer que les informations ne sont pas véhiculées uniquement par la couleur, mais également que l'alternative est accessible.

Cette première partie de la thématique « Couleurs » est brève, simple en apparence, mais peut parfois nécessiter de créer de véritables versions alternatives, comme nous l'avons vu à travers l'exemple de Météo France. C'est la condition sine qua non pour n'exclure aucun internaute.

Le second volet de cette thématique concerne les contrastes.

Un contraste « suffisant »

Imaginez... Formidable, vous refaites votre charte graphique ! À la fois très classe, avec ses nuances de gris, et dynamique, avec ses tons orange et jaune, vous êtes certain que tout le monde va aimer ! « Tout le monde », vraiment ? Qu'en est-il des personnes ayant des troubles de la vision, des personnes âgées dont la vue a baissé, et de tous ces utilisateurs de smartphones et tablettes qui ont pris l'habitude de surfer en extérieur, là où la luminosité rend les faibles contrastes peu perceptibles ?

À NOTER Halte aux textes illisibles !

Ce problème de textes rendus peu lisibles en raison d'un faible contraste est malheureusement récurrent sur le Web et interpelle les utilisateurs et faiseurs de Web, bien au-delà des experts accessibilité.

Un exemple ? Cette initiative appelant à lutter contre les faibles contrastes (en anglais) :

► <http://contrastrebellion.com>

Aïe, patatras ! Vous aviez oublié de prendre en compte les critères d'accessibilité...

Pas de panique ! Il n'est jamais trop tard pour bien faire. Voyons en détail de quoi il retourne.

Les contrastes de couleurs sont traités par les critères 3.3 et 3.4 du référentiel Accessi-Web. Les deux indiquent qu'il faut veiller au contraste entre les couleurs de premier et d'arrière-plan. Mais alors que le critère 3.3, de niveau Argent, demande à ce qu'il y ait un contraste « suffisant », le critère 3.4, de niveau Or, parle quant à lui de contraste « amélioré ».

Sous AccessiWeb 1, le niveau du contraste était évalué « à l'œil », de façon subjective, ce qui posait des problèmes d'interprétation qui pouvait différer selon la personne qui évaluait. Avec les WCAG 2, la WAI a mis au point une formule pour calculer de façon objective la qualité du contraste entre une couleur de premier plan et une couleur d'arrière-plan. Cette formule prend en compte divers paramètres, comme la taille des caractères, leur graisse et la luminosité de chaque couleur.

DÉBAT Pertinence relative de la formule de calcul des WCAG 2

Avoir une formule mathématique qui permet d'évaluer le bon contraste, ça semble magique ! Trop beau pour être vrai ? Eh bien, un peu, oui...

S'il a été jugé préférable d'avoir recours à un calcul objectif plutôt que de faire reposer l'évaluation sur une expertise purement humaine, la formule n'est pas parfaite.

On remarquera ainsi certains cas où le validateur automatique recale un couple de couleurs qui, pourtant, semble très lisible. Et inversement, l'outil peut être amené à valider des contrastes qui mériteraient d'être améliorés. Sans compter que la notion de fatigue sur écran n'est pas prise en compte. On sait qu'une trop grande luminosité, liée à un contraste excessif, constitue un handicap pour les personnes dyslexiques ou hypersensibles à la lumière.

Retrouvez un article très complet de Nicolas Hoffmann sur le site d'OpenWeb :

► <http://ur1.ca/87kx2>^a

Et pour les plus aventureux qui souhaiteraient connaître la formule de contraste, reportez-vous au glossaire disponible sur le site des WCAG 2 :

► <http://ur1.ca/9ycth>^b

a. http://openweb.eu.org/articles/accessibilite_contrastes_textes_sites

b. <http://www.w3.org/Translations/WCAG20-fr/#contrast-ratiodef>

Grâce à cette formule, et pour notre plus grand bonheur, des outils ont pu être mis au point pour évaluer de façon automatique si le texte est suffisamment contrasté selon le niveau AA ou AAA des WCAG 2. Il n'est donc pas nécessaire de se torturer à faire de savants calculs pour connaître la réponse aux questions posées par les critères 3.3 et 3.4 du référentiel AccessiWeb.

S'il est essentiel de veiller à proposer un contraste suffisant, autant le dire tout de suite : l'exigence du critère 3.4 est telle qu'il est très difficile de le respecter. Heureusement, il s'agit d'un critère de niveau Or et AccessiWeb parle bien de contraste « amélioré ».

L'objectif ici va donc être, en termes d'accessibilité, de viser le respect du critère 3.3, de niveau Argent.

ATTENTION Non aux « versions accessibles » affreuses !

Il arrive que certains éditeurs de site, conscients du manque de contraste, prévoient une version accessible plus contrastée. Or, au lieu de seulement foncer les couleurs posant problème, dans le respect de leur identité visuelle, ils proposent une version sans CSS, donc sans couleur ni mise en page. L'utilisateur se voit alors imposer un choix entre voir un joli site coloré mais sans pouvoir déchiffrer ce qui est écrit, ou lire le contenu, mais sur une version affreuse du site.

Une version accessible sans CSS, si elle respecte les autres critères d'accessibilité, à savoir permettre l'accès aux mêmes contenus et fonctionnalités, aussi à jour que la version non accessible, pourra être considérée comme respectant les WCAG, le RGAA et AccessiWeb. Mais quelle insulte faite aux internautes !

Car si les critères sont respectés à la lettre, qu'en est-il alors de l'esprit de l'accessibilité web qui s'inscrit dans un objectif de conception universelle ? (Voir l'aparté à ce sujet dans le chapitre 1, page 14.)

Si vous vous intéressez à l'accessibilité web, au-delà de simplement respecter la loi, pensez à respecter également vos internautes en leur proposant un contenu de qualité, sous une forme accessible et agréable pour tous !

Les limites de la seule prise en compte des contrastes

Nous l'avons rapidement évoqué dans l'encadré précédent sur la pertinence relative de la formule de calcul, mais il est important de revenir sur ce point. Le référentiel AccessiWeb, au même titre que le RGAA ou les WCAG dont ils sont tous deux issus, ne sont que des outils. Des outils bien pratiques, indispensables même, mais également imparfaits et à utiliser avec discernement.

Le fait qu'il y ait un critère sur les contrastes avec une formule permettant de bien distinguer si le contraste est suffisant est un bon exemple. Car s'il prend en compte les personnes ayant besoin d'un contraste élevé pour percevoir le texte, il laisse totalement de côté ceux pour qui un contraste trop élevé est synonyme de brouillage de la vue.

Ne vous est-il jamais arrivé, quand vous êtes très fatigué, de sortir dans la rue le soir et de devoir plisser les yeux pour lire le texte inscrit sur les panneaux lumineux ? De même, si vous travaillez longtemps dans une pièce sombre sur un ordinateur dont la

luminosité est au maximum, il y a des chances pour que votre vue se brouille ou que le texte sur fond blanc commence à vous faire mal à la tête. C'est d'ailleurs pour cette raison que les liseuses de livres électroniques, dont on se sert pour lire de longs textes, sont dotées d'un écran qui n'est pas rétro-éclairé.

Or, cette gêne que tout le monde peut ressentir un jour ou l'autre, peut représenter un véritable handicap pour certaines personnes, dyslexiques ou hypersensibles à la lumière. Pourtant, en incitant à faire de forts contrastes, les WCAG 2 ne prennent pas du tout en compte ce type de handicap.

Dès lors, comment faire ? Notre conseil sera de chercher le meilleur compromis, dans le cadre du niveau AA (Argent) et de ne surtout pas chercher à atteindre le niveau AAA (Or). En résumé, respecter l'objectif de contraste suffisant, mais sans oublier qu'un texte noir sur un léger fond gris plutôt qu'un fond blanc permettra à certains de mieux naviguer sur vos pages...

OUTILS Vérifier le contraste des couleurs selon la formule des WCAG 2

Nous allons aborder ici trois outils, pour trois contextes d'utilisation différents.

Si vous souhaitez évaluer les contrastes des couleurs sur une page web déjà codée en HTML/CSS, l'extension Firefox **WCAG Contrast Checker** (figure 6-16) est faite pour vous. Une fois installée, il suffit d'ouvrir la page à tester et de cliquer sur le bouton de l'extension indiqué par un cercle coloré.

Un panneau s'ouvre alors avec deux onglets : le premier, intitulé *Élément*, vous permet d'effectuer des tests sur des éléments précis de la page : le second, intitulé *Document*, vous permet en un clic de vérifier les contrastes sur toute la page. Évidemment, vous devez spécifier que l'évaluation doit être réalisée à partir des WCAG 2 et non de la version 1, également proposée. Nous vous conseillons également de tester plutôt le niveau AA, même si le niveau AAA est disponible.

Les éléments analysés sont listés et apparaissent en rouge quand les critères d'accessibilité sont invalidés, et en bleu quand ils sont validés. Vous noterez que l'information est donnée par la couleur, et seulement par la couleur... Une extension très pratique donc, mais pas très accessible !

Pour télécharger WCAG Contrast Checker, rendez-vous à l'adresse suivante :

► <http://ur1.ca/9yl3m^a>

À présent, vous procédez à la refonte de votre site et vous souhaitez analyser plus finement les couleurs de votre charte actuelle, afin d'effectuer des tests pour améliorer les contrastes. Pour cela, l'outil en ligne (en anglais) **Color Contrast Check** vous sera utile (figure 6-17).

Il suffit d'entrer le code hexadécimal des couleurs de premier et d'arrière-plan dans les champs prévus à cet effet afin d'obtenir des informations sur les contrastes, la conformité aux WCAG 2, mais aussi sur la luminosité. Vous pouvez également modifier à loisir les valeurs des couleurs pour tester ce qui convient le mieux.

La page de Color Contrast Check se trouve à l'adresse suivante :

► <http://ur1.ca/9yl5l^b>

Enfin, si vous n'avez pas encore de maquette HTML à tester, mais que vous travaillez en phase de conception avec un logiciel de retouche d'images, vous souhaitez sans doute tester vos couleurs en amont pour savoir si elles sont accessibles afin de ne pas avoir à tout modifier plus tard. Vous pourrez alors utiliser l'outil en ligne **Cercle Chromatique Accessible** (figure 6-18, page suivante). Vraiment très pratique, il permet de choisir dans une roue chromatique les couleurs de premier plan, par exemple, et de bouger le curseur sur la roue pour visualiser si la couleur de fond permet de valider l'accessibilité ou pas. Si la couleur choisie est suffisamment contrastée, le terme « OK » apparaît à droite de l'écran. Dans le cas contraire, rien ne s'affiche. Découvrez l'outil Cercle Chromatique Accessible en ligne :

► <http://ur1.ca/9yl7i>

- <https://addons.mozilla.org/fr/firefox/addon/wcag-contrast-checker/>
- http://www.snook.ca/technical/colour_contrast/colour.html
- <http://gmazzocato.altervista.org/fr/colorwheel/wheel.php>

Figure 6-16

Article sur les couleurs de Nicolas Hoffmann publié sur le site OpenWeb et analysé par WCAG Contrast Checker



Figure 6-17

L'outil en ligne Color Contrast Check

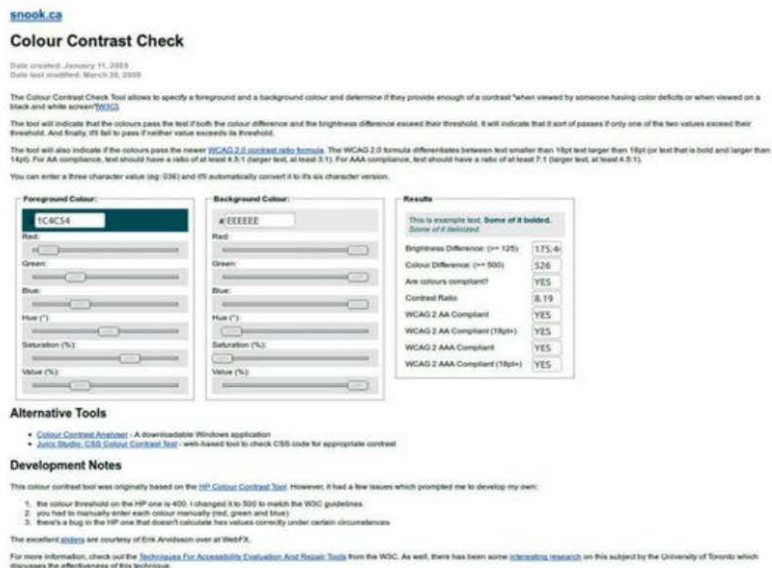
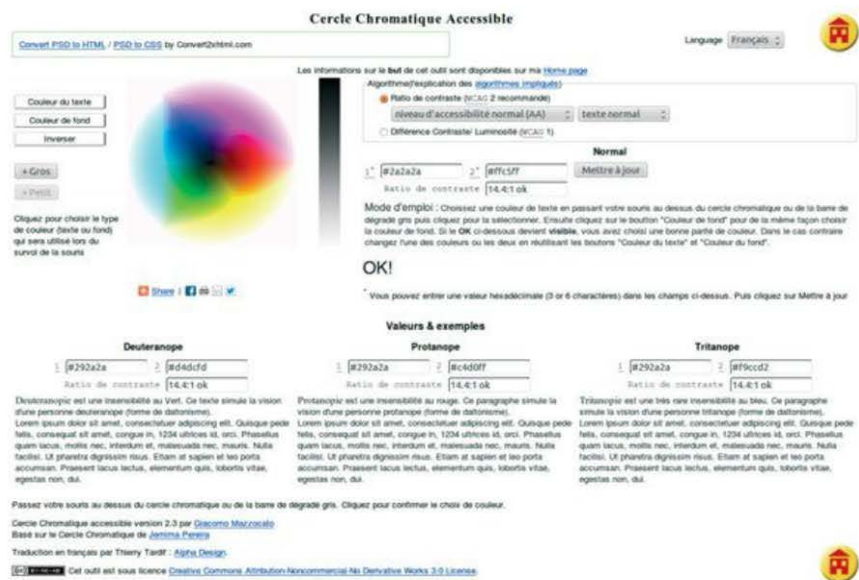


Figure 6–18
L’outil Cercle Chromatique Accessible



Multimédia

(thématique AccessiWeb 4)

« Donner si nécessaire à chaque média temporel une transcription textuelle, des sous-titres synchronisés et une audio-description synchronisée pertinents. Donner à chaque média non temporel une alternative textuelle pertinente. Rendre possible le contrôle de la consultation de chaque média temporel et non temporel au clavier et s'assurer de leur compatibilité avec les technologies d'assistance. »

<http://ur1.ca/aelbk>¹

Aujourd'hui, communiquer sur le Web ne consiste plus seulement à écrire des textes égayés par des images, mais également à publier des vidéos, enrichir ses pages d'animations pour les rendre plus attractives, proposer des jeux ou des visites virtuelles de lieux ouverts au public... Bref, les modes de communication ont évolué et le multimédia a envahi les pages web grâce à des moyens facilitant leur publication.

Outre l'aspect « moderne » et « dynamique » que cela apporte, c'est aussi une façon pour vos utilisateurs de prendre connaissance de vos contenus autrement. Et c'est une très bonne chose pour certaines personnes ayant, par exemple, du mal avec l'écrit (dyslexie, troubles de la concentration) ou se trouvant en situation de handicap mental, car elles comprendront sans doute mieux une animation qu'un texte. Cela offre aussi la possibilité de faire des traductions en langue des signes française (LSF).

Le multimédia fourmille donc de points positifs, tant pour votre communication que pour aider certains utilisateurs en situation de handicap à accéder à vos contenus. Mais la communication multimédia, pour être vraiment accessible à tous, nécessite de respecter des règles assez nombreuses. Car si les vidéos, animations et autres éléments multimédias facilitent l'accès au Web pour certains, ils peuvent aussi, s'ils ne sont pas rendus accessibles, être un barrage supplémentaire pour d'autres personnes :

1. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#multimedia

- les personnes sourdes : par exemple, un média audio qui n'est pas accompagné de sous-titre ni d'une version texte leur sera totalement inaccessible ;
- les personnes aveugles : comme pour les images, tout ce qui est visuel doit avoir une alternative sous forme de texte ; et dans le cas d'une vidéo, tout ce qui est communiqué uniquement visuellement, en dehors des dialogues, par exemple, ne sera pas perçu ;
- les personnes ne pouvant naviguer qu'au clavier : il peut s'agir de personnes en situation de handicap moteur ou de personnes aveugles. Si le lecteur multimédia n'est pas accessible au clavier, et si les boutons de contrôle du média (boutons Lecture, Pause...) n'ont pas d'alternative textuelle, le média ne sera pas accessible.

Ainsi, le multimédia ouvre de nouvelles opportunités pour communiquer autrement sur le Web, ce qui est une chance. Mais si l'on ne veut pas que cette opportunité se transforme en frein pour certains utilisateurs, il faut penser à respecter quelques règles d'accessibilité, que nous allons à présent étudier.

INFO Critères de la thématique « Multimédia »

La thématique « Multimédia » contient vingt-deux critères, répartis en quinze critères de niveau Bronze, quatre critères de niveau Argent et sept critères de niveau Or.

Cette thématique est la plus longue, les enjeux posés par le multimédia accessible étant assez complexes. Nous traiterons de tous les critères présents dans la thématique, à l'exception du critère 4.19, de niveau Or, relatif à l'évaluation d'un niveau sonore suffisant, ce qui est en pratique presque impossible à tester.

Pour plus de détails sur le critère 4.19, consultez la version déployée du référentiel AccessiWeb 2.2 :

► <http://ur1.ca/9z4n5a>

a. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-4-19

Vocabulaire : média temporel, média non temporel, synchronisé ou non, préenregistré ou en direct

Outre le fait que cette thématique « Multimédia » est la plus longue, c'est aussi la plus difficile à lire et à comprendre si l'on ne maîtrise pas les concepts de base.

Média temporel versus média non temporel

Il s'agit d'un objet multimédia (par exemple, inséré via la balise `<object>`) et qui peut soit se dérouler dans le temps, soit être statique.

Si le média a une durée, avec un début et une fin (comme un morceau de musique, une vidéo, une animation), il s'agit d'un média temporel.

Si le média est statique (comme une publicité en Flash, une visite virtuelle dans laquelle l'internaute peut se déplacer, une carte interactive, un diaporama fixe), c'est-à-dire qu'il ne se déroule pas dans le temps, il s'agit d'un média non temporel.

À NOTER WCAG 2 ne parle pas de médias non temporels

Cette distinction entre média temporel et média non temporel est propre à AccessiWeb, les WCAG 2 ne définissant que la notion de média temporel.

Média synchronisé ou non synchronisé

Les WCAG 2 distinguent quatre types de médias (<http://ur1.ca/9ymbw²>) :

- 1** seulement audio (média non synchronisé) ;
- 2** seulement vidéo (média non synchronisé) ;
- 3** audio et vidéo (média synchronisé) ;
- 4** audio et/ou vidéo combiné avec de l'interactivité (média synchronisé).

Ainsi, vous pouvez avoir une animation sans son ou une interview radio sans image. Les flux vidéo et audio sont utilisés séparément. Il s'agit de médias non synchronisés.

Mais il arrive aussi que vous ayez une vidéo avec du son ou une animation mêlant des images, de la musique et du texte audio. Les flux images et son sont alors synchronisés.

Enfin, il peut y avoir la combinaison d'un flux audio et/ou vidéo avec d'autres flux permettant une interaction en cours de lecture du média. C'est le cas, par exemple, d'une vidéo dans laquelle apparaît un lien pour proposer à l'internaute de cliquer tout de suite pour en savoir plus. L'interactivité doit avoir lieu à un instant particulier. Avoir une transcription textuelle de la forme « Pour plus d'informations, cliquer maintenant » ne serait pas très utile puisque l'internaute ne pourrait pas savoir avec précision à quelle moment se réfère le « maintenant ». En conséquence, des sous-titres synchronisés seront nécessaires.

En résumé, un média synchronisé est un média qui allie son et images ou bien l'un *et/ou* l'autre, associé à de l'interactivité.

Média préenregistré ou en direct

Un média préenregistré est un média dont le fichier est enregistré sur votre disque dur (ou ailleurs) et que vous envoyez sur le Web. Vous pouvez le regarder indépendamment sur votre ordinateur, sans être connecté.

2. <http://www.w3.org/Translations/NOTE-UNDERSTANDING-WCAG20-fr/media-equiv.html>

À l'opposé, un média en direct est un média en lecture continue, c'est-à-dire un média qui retransmet un événement à mesure qu'il se déroule. Cela passe par une diffusion en mode continu de l'événement. On parle aussi de *streaming*.

Transcription et sous-titrage : écrire ce qui ne peut être entendu

Nous l'avons évoqué en introduction, la thématique « Multimédia » est pensée, entre autres, pour les personnes sourdes et malentendantes.

Essayez de comprendre un reportage sans le son. Difficile, voire impossible, n'est-ce pas ? Véhiculer un contenu uniquement par ce biais revient donc à exclure toutes les personnes ne pouvant pas entendre.

La solution ? Fournir, en plus de la vidéo, une autre façon d'accéder au contenu.

Pour cela, il existe deux moyens.

- **La transcription textuelle** : il s'agit de reproduire sous forme de texte tout ce qui se passe sur le média (audio et/ou vidéo). C'est comme si vous vouliez rendre disponible le média en question pour quelqu'un qui n'a pas Internet. Vous lui envoyez une version texte, comprenant bien sûr les dialogues et voix off s'il y en a, et qui décrit également tous les éléments importants, visuels (comme une personne qui hoche la tête) et auditifs (une porte qui claque, par exemple).

La transcription textuelle offre une alternative accessible, non seulement pour les personnes ne pouvant entendre la vidéo, mais également pour les personnes ne pouvant pas la voir. Et cela rend possible du même coup le référencement du contenu de la vidéo par les moteurs de recherche.

Un autre avantage de la transcription textuelle, au-delà de l'accessibilité, est qu'elle permet de parcourir rapidement le texte avant de savoir si on a envie de prendre le temps de lire la vidéo. Contrairement à un média temporel, un texte ne présente pas de contrainte de durée et permet d'effectuer une recherche par mot-clé, grâce au raccourci clavier **Ctrl+F** du navigateur. Ainsi, les contenus texte (dont les transcriptions) sont souvent préférés par les internautes pressés.

- **Le sous-titrage pour sourds et malentendants** : une version texte, c'est bien, mais cela change totalement l'expérience utilisateur. On ne vit pas la même chose quand on lit un texte que quand on lit un média comme une vidéo. Les images peuvent véhiculer des émotions et la sensation est différente quand elle est véhiculée par l'image ou par des mots écrits. C'est là tout l'intérêt du sous-titrage. Synchronisé avec le média, il permet à l'utilisateur de vivre en temps réel ce qui se déroule sur le média.

VOCABULAIRE Caption & subtitle

Si, en français, le même mot de « sous-titrage » est utilisé, il faut noter que l'anglais distingue le *captioning*, sous-titrage pour sourds et malentendants, du *subtitling*, sous-titrage pour de la traduction. Les deux sont semblables sur le plan technique du calage (synchronisation) et de l'incrustation, mais le travail est très différent du point de vue de la rédaction.

En effet, l'objectif n'est pas le même et il est important d'avoir recours à des professionnels connaissant le sous-titrage pour sourds et malentendants dans le cadre d'une mise en accessibilité, pour que tous les effets sonores ayant de l'importance trouvent une transcription écrite dans le cadre du sous-titrage.

Notons que si le sous-titrage est rendu obligatoire par le critère 4.3, de niveau Bronze, la transcription n'est quant à elle obligatoire qu'au niveau Or (critère 4.13). Au niveau Bronze, le choix est offert entre transcription et audiodescription (critère 4.1).

Proposer une transcription et un sous-titrage, c'est bien. Mais il faut aussi que ces alternatives soient... pertinentes ! C'est même essentiel. Et cette obligation est également présente pour les médias diffusés en direct (critères 4.5 et 4.6, de niveau Argent).

INFO Sous-titrage en direct

Sous-titrer un média en direct, surtout lorsqu'il s'agit d'une vidéo comme un journal télévisé ou une interview, est un exercice complexe qui nécessite un dispositif matériel et humain sophistiqué.

La plupart des chaînes de télévision utilisent des logiciels de reconnaissance vocale. Une première personne, appelée le « perroquet », est chargée de répéter ce qui est dit en direct. Cette répétition dans un cadre calme permet à la fois d'insérer les signes de ponctuation et une meilleure compréhension par le logiciel que si la reconnaissance était faite directement à la source, avec tous les inconvénients que l'on peut imaginer (bruits de fond, interlocuteur avec un fort accent...). Une deuxième personne, le correcteur, corrige ensuite les fautes d'orthographe générées inévitablement par le logiciel. Parfois, une troisième personne fait une dernière relecture avant de valider le texte. Le correcteur peut également être validateur. L'ensemble du processus se passe en quelques secondes pour éviter un décalage trop grand entre le direct et le sous-titrage.

Trois personnes, un logiciel spécialisé, et pourtant encore des erreurs... Eh oui, car malgré les moyens mis en place, le sous-titrage en direct reste un exercice complexe. Il relève pourtant du niveau Argent, car sans cela, l'accès à l'information est réellement perdu. Si les moyens vous manquent, vous pouvez décider de sous-titrer la vidéo après diffusion, ce qui est beaucoup plus abordable.

Pour avoir une idée en images de ce que cela donne, vous pouvez consulter un reportage sur Dailymotion qui présente les coulisses du sous-titrage pour le journal télévisé de France 3 (la vidéo est sous-titrée).

► <http://ur1.ca/9z26m^a>

a. http://www.dailymotion.com/video/xazms7_le-sous-titrage-des-jt-de-france-3_news

Prenons un exemple concret. Vous suivez un débat politique retransmis en direct sur le site web de votre chaîne favorite. Le sous-titrage est prévu, c'est une chance ! Vous n'avez pas de télévision à disposition et le son ne fonctionne plus sur votre vieil ordinateur de secours (l'autre est en panne...). Le débat porte sur le mariage et l'adoption par des personnes de même sexe : faut-il ou non l'autoriser et l'inscrire dans la loi ? Au début, il vous semble que le candidat du parti de gauche y est favorable, tandis que le candidat de droite s'y oppose. Puis, à mesure que le débat avance, vous n'y comprenez plus rien ! Les sous-titres se décalent légèrement et s'inscrivent sous le mauvais orateur. Imaginez la confusion !

Ce genre d'incident est malheureusement souvent une réalité. Le site Médias sous-titrés relate ainsi le problème de décalage survenu lors de la retransmission en direct d'une allocution de l'ancien président Nicolas Sarkozy, le 28 octobre 2011. Le décalage des sous-titres était tel qu'il était impossible de savoir si ce qui apparaissait à l'écran était une déclaration du chef de l'État ou un commentaire d'un journaliste...

SOURCE Catastrophe durant l'allocution présidentielle de Nicolas Sarkozy

Vous pouvez lire l'article qui relate cet incident sur le site de Médias sous-titrés à l'adresse suivante :

► <http://ur1.ca/9yvbja>^a

a. <http://www.medias-soustitres.com/television/actualites/Catastrophe-durant-l-allocution>

Et les sourds qui ne savent pas lire ?

Les critères relatifs à une alternative en langue des signes française (LSF) sont tous deux de niveau Or (critères 4.9 et 4.10). Est-ce à dire que ce n'est pas grave si ces personnes ne peuvent pas accéder aux contenus ? Non, bien sûr. Mais réaliser une vidéo en langue des signes, sur un support tel que le Web prévu pour véhiculer au départ textes et images, demande beaucoup de moyens : des interprètes compétents, un tournage spécifique, un montage et une publication qui ralentissent la diffusion de la vidéo. Il n'existe pas encore de solution fiable permettant une traduction automatique en LSF. D'autant qu'au coût de production d'une vidéo en LSF, il faut prévoir celui de l'évaluation d'accessibilité qui implique de vérifier que la traduction est... pertinente (critère 4.10). L'investissement tant humain que financier est donc considéré comme trop important compte tenu des moyens technologiques d'aujourd'hui pour rendre obligatoire la diffusion de tous les contenus en LSF. C'est pourquoi il s'agit de critères de niveau Or.

CANULAR Une interprète en langue des signes inventant son texte au JT

Un matin, le réveil s'allume et se branche sur France Info. Une annonce indique qu'une interprète en langue des signes sur le journal télévisé de la BBC a été licenciée pour avoir diffusé de fausses informations. Au lieu de traduire ce que disaient les journalistes, elle inventait des histoires, comme celle disant que des zombies se seraient réveillés suite à l'accident nucléaire de Fukushima. Cette histoire a été largement reprise par les médias français, avant que tout le monde ne se rende compte qu'il s'agissait d'un canular. Cette fausse information avait été diffusée sur le site satirique *The Poke* en août 2011.

Il est presque dommage pour notre propos que cette histoire ne soit pas vraie. Outre la leçon que tout journaliste devrait en tirer (vérifier ses sources avant de diffuser n'importe quoi...), cela montre aussi qu'on ne peut s'exonérer d'une vérification de la pertinence tant des traductions que des transcriptions ou sous-titrages... Même s'il est vrai que, dans la pratique, cet exercice reste difficile.

Plus d'informations (en anglais) sur le site *The Poke* :

► <http://ur1.ca/9yvbw>^a

a. <http://www.thepoke.co.uk/2011/08/17/bbc-sign-language-interpreter-sacked-for-changing-the-news/>

Ainsi, la LSF est bien prévue tant par AccessiWeb que par le RGAA ou les WCAG, mais au niveau Or uniquement. En revanche, le sous-titrage est, lui, obligatoire.

IDÉE REÇUE « Le sous-titrage vidéo coûte trop cher ! »

Le sous-titrage de vidéo demande du temps et de l'argent, surtout si on le fait soi-même et sans formation. Mais si vous externalisez et confiez le travail à des personnes compétentes, le budget à allouer n'est pas du tout inatteignable. Il suffit pour cela de l'intégrer en amont dans le budget relatif à la vidéo (préparation, tournage, montage, diffusion... et sous-titrage). Par ailleurs, si vous décidez de vous former pour le faire vous-même, vous pouvez très bien vous servir de vos notes pour la préparation. En général, il y a déjà une trace écrite qui a permis de préparer la vidéo. Parfois, tout est écrit d'avance. Il suffit alors de reprendre le texte et de le caler pour qu'il soit synchronisé avec la vidéo.

À titre indicatif, selon une petite étude personnelle, il faut compter entre 10 et 12 euros HT la minute de sous-titrage pour un contrat annuel sur des vidéos de 3 à 5 minutes, une fois par semaine. Cela comprend le calage de la vidéo (la synchronisation) et la fourniture d'un fichier de sous-titres (SRT) que vous pouvez ajouter simplement sur des plates-formes telles que Dailymotion ou YouTube.

Si le coût du sous-titrage est intégré en amont dans le budget, ce n'est donc pas forcément un problème. Et le bénéfice est immédiat, tant pour vos utilisateurs (sourds, mais aussi ne maîtrisant pas le français ou n'ayant pas de son) que pour votre référencement.

Audiodescription : dire ce qui ne peut être vu

Certaines personnes n'ont aucun mal à entendre les médias, mais ne peuvent pas les voir, ce qui peut poser problème dans le cas d'une vidéo. Si une question est posée et que la personne répond en hochant ou en secouant la tête, par exemple, ou si certains bruits surviennent pour indiquer une action (une explosion, un grincement, un craquement...) sans autre indice pour savoir ce qui se passe, on perd de l'information et on peut vite ne plus rien comprendre à ce qui se déroule.

Imaginez-vous en train de regarder certaines scènes de Star Wars sans images. Même si vous connaissez cette scène par cœur, si on vous la passe hors contexte et sans images, comment sauriez-vous lequel du sabre laser « zzzem zzzem » est en train de remporter la bataille ?

L'audiodescription vise à combler ce vide en ajoutant une voix qui va décrire les informations importantes qui ne sont véhiculées que par le visuel, afin que les auditeurs ne soient pas perdus. Dans notre exemple de Star Wars, cela reviendrait, pendant les scènes de batailles, à décrire celui qui a la main : « soudain il tombe, puis esquive un coup (bruit de fracas), puis revient sur ses jambes et finit par l'emporter ».

INFO Coût de l'audiodescription

Le site de l'Association Valentin Haüy (AVH) indique : « Le tarif moyen de l'audiodescription varie aujourd'hui, en France, entre 55 et 70 euros par minute (soit entre 5 000 et 6 300 euros pour une fiction de 90 minutes). »

Mais les vidéos diffusées sur le Web sont en général beaucoup moins longues... Ainsi, pour une vidéo de 5 minutes, cela reviendrait entre 275 et 350 euros selon les tarifs moyens donnés par l'AVH.

► <http://ur1.ca/9z2ut>^a

a. http://www.avh.asso.fr/rubriques/audiovision/tout_savoit_audiovision.php

Si les vidéos sous-titrées sur le Web sont rares, que dire des vidéos audiodécrites ? Cette technique est encore plus rarement utilisée. Et pourtant, facultative au niveau Bronze (le critère 4.1 laissant le choix entre audiodescription et transcription textuelle), l'audiodescription est rendue obligatoire au niveau Argent (critère 4.7).

Bien, vous avez compris les enjeux et travaillé à rendre vos œuvres accessibles au plus grand nombre. La vidéo est accessible pour les sourds grâce à du sous-titrage, elle est également audiodécrite. Mais encore faut-il maintenant que le support qui permet sa diffusion sur la page web soit lui-même accessible.

Un lecteur accessible

Description synthétique du média

L'accessibilité du lecteur de média temporel ou du média non temporel est cadrée par les critères 4.15 à 4.22, tous de niveau Bronze à l'exception du critère 4.19, de niveau Or.

La première des choses à faire pour un lecteur de média temporel consiste à décrire de façon synthétique le contenu de ce média en alternative. S'agit-il d'une vidéo, d'un morceau audio ? Quel en est le contenu ? Ces informations seront restituées pour permettre à l'internaute de savoir s'il veut entrer dans l'objet. C'est également le texte qui sera affiché si l'internaute n'a pas la technologie installée nécessaire à la lecture du média (Flash, la plupart du temps).

COMPRENDRE Flash et l'accessibilité

Flash n'est pas du Web. Si c'est une évidence pour certains, cela peut paraître une incongruité pour d'autres, car c'est bien sur le Web qu'on trouve du Flash. Il n'est donc pas inutile de rappeler que les « objets » Flash sont « embarqués » sur des pages web. D'où l'utilisation de balises `<object>` et `<embed>` quand on veut insérer du Flash sur des pages web.

À une époque où le Web était encore techniquement très limité, les possibilités offertes par ces technologies Flash et autres ouvraient un champ immense aux créatifs pour leur permettre d'insérer de l'interactivité ou du multimédia sur les pages.

Or, alors que le Web repose sur des technologies ouvertes pour permettre la meilleure interopérabilité possible et une diffusion universelle, les technologies comme Flash de la société Adobe, ou Silverlight de Microsoft, sont propriétaires, fermées et imposent à l'internaute d'avoir sur son poste des greffons propriétaires pour pouvoir les lire.

Par ailleurs, techniquement, Adobe a travaillé pour permettre de rendre sa technologie accessible. Mais il s'agit d'un travail effectué après la création du langage ActionScript, et la complexité de mise en œuvre de l'accessibilité d'un objet Flash est telle que rares sont ceux qui, aujourd'hui, ont la compétence requise pour y arriver. Quand c'est seulement possible, ce qui n'est pas toujours le cas. Un site entièrement en Flash requerra toujours une alternative en HTML pour être réputé accessible, ce qui revient à maintenir deux sites. Est-ce vraiment raisonnable ?

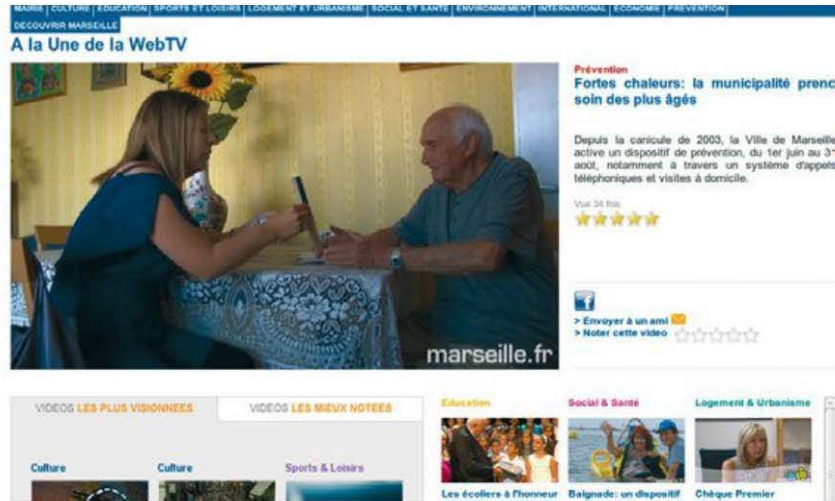
Un site entièrement conçu en Flash ne sera pas accessible, mais on peut, dans une certaine mesure, rendre du Flash accessible (notamment les lecteurs médias). Quoiqu'il en soit, rien ne vaut des technologies ouvertes en matière d'accessibilité. Et la bonne nouvelle, c'est qu'avec JavaScript et HTML5, tout est désormais possible !

Contrairement à l'habitude d'AccessiWeb, la présence de la description et sa pertinence sont évaluées en un seul critère, le critère 4.15.

La description peut être donnée par tout moyen : que ce soit via un lien adjacent, entre les balises `<object>` et `</object>` ou encore au sein de balises `<noembed>`.

Figure 7-1

Web TV de la ville de Marseille
(<http://ur1.ca/9zr5h2>)



a. <http://www.marseilletv.fr/sitevideo/jsp/site/Portal.jsp>

Le code de la vidéo à la Une (figure 7-1) est le suivant :

```
<object id="flashfirebug_1345451342499" width="640" height="360"
data="flash/flash_video.swf" type="application/x-shockwave-flash">
<param value="lecteur=flash/MojaveExternalAll.swf&maVideo=http://
www2.mairie-marseille.fr/videos/
social_120817_canicule.flv&fileVar=document&idVar=1502&id_attribute=73&
autostart=0&comptageVideo=jsp/site/plugins/document/
manageVisites.jsp?document_id=1502" name="flashvars">
<param value="true" name="loop">
<param value="false" name="menu">
<param value="transparent" name="wmode">
<param value="true" name="allowFullScreen">
<p>Video</p>
<param name="allowScriptAccess" value="always">
<param name="AllowNetworking" value="all">
</object>
```

On remarque qu'un texte de remplacement est bien prévu entre les balises `<object>`, mais le terme « Video » n'est pas pertinent puisqu'il ne s'agit en aucun cas d'une description synthétique du contenu du média.

Pour valider le critère 4.15, il aurait plutôt fallu corriger le code de la façon suivante :

```
<object id="flashfirebug_1345451342499" width="640" height="360"
data="flash/flash_video.swf" type="application/x-shockwave-flash">
<param value="lecteur=flash/MojaveExternalAll.swf&maVideo=http://
www2.mairie-marseille.fr/videos/
social_120817_canicule.flv&fileVar=document&idVar=1502&id_attribute=73&
autostart=0&comptageVideo=jsp/site/plugins/document/
manageVisites.jsp?document_id=1502" name="flashvars">
<param value="true" name="loop">
<param value="false" name="menu">
<!-- Suppression du mode transparent, équivalent d'un CSS display:none,
invisible pour les lecteurs d'écran (voir pages suivante) -->
<param value="true" name="allowFullScreen">
<!-- On fait une description synthétique du contenu de la vidéo -->
<p>Reportage sur le dispositif canicule à Marseille : témoignage vidéo
d'un vieux monsieur</p>
<param name="allowScriptAccess" value="always">
<param name="AllowNetworking" value="all">
</object>
```

Quand il ne s'agit pas d'un média temporel, mais d'un média non temporel, comme une carte interactive, le critère 4.16 nous indique qu'il faut toujours prévoir une alternative accessible.

Là encore, cette alternative peut être apportée par tout moyen. Les cas sont listés dans le détail du critère (<http://ur1.ca/9zcu2>³). Évidemment, cette alternative doit être... pertinente.

BONNE PRATIQUE Une alternative en dehors de l'objet

Les meilleures alternatives sont celles qui sont disponibles à l'extérieur de la balise `<object>`, via un lien adjacent, ce qui évite à l'utilisateur qui ne pourrait utiliser les médias ou les animations de désactiver Flash pour accéder à l'alternative.

Un média contrôlable par l'utilisateur

Pour qu'un lecteur média soit accessible, il doit aussi être contrôlable par l'utilisateur. Cela signifie que le son du média doit être contrôlable indépendamment du son de l'ordinateur. L'internaute doit pouvoir augmenter, baisser ou couper le son, sans que cela ait de conséquences sur le son de son ordinateur.

3. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-4-16

Par ailleurs, si le son se lance automatiquement, soit la durée est inférieure à trois secondes avant que cela ne s'arrête, et ce n'est pas un problème ; soit le son dure plus de trois secondes et il faut permettre à l'utilisateur de le couper.

Un lecteur contrôlable, c'est également un lecteur au sein duquel on peut naviguer au clavier et à la souris. Notons que le contrôle au clavier s'effectue grâce aux touches combinées **Tab+Entrée**, sauf pour les contrôles complexes.

Nous parlons de lecteur de média en pensant à l'audio et à la vidéo, mais c'est également valable pour les médias non temporels, telle qu'une carte interactive, au sein de laquelle l'utilisateur doit pouvoir naviguer aussi bien au clavier qu'à la souris.

Un média « compatible avec les technologies d'assistance »

Enfin, un média accessible doit être « compatible avec les technologies d'assistance » (critère 4.22). L'intitulé de ce critère doit vous faire penser à la notion très vague de « compatible avec l'accessibilité » des WCAG 2, non ?

Voyons donc en détail ce qu'il dit précisément dans sa version déployée (avec le test qui l'accompagne) :

« **Critère 4.22 [Bronze]** Chaque média temporel et non temporel est-il compatible avec les technologies d'assistance ?

Test 4.22.1 : Chaque média temporel et non temporel inséré via une balise **Object**, **Applet** ou **Embed** vérifie-t-il une de ces conditions ?

Le nom, le rôle, la valeur, le paramétrage et les changements d'états des composants d'interfaces sont accessibles aux aides techniques via une API d'accessibilité.

Une alternative compatible avec une API d'accessibilité permet d'accéder aux mêmes fonctionnalités. »

Prenons l'exemple d'un lecteur vidéo sur lequel un bouton **Lecture** se transforme en bouton **Pause** quand il est activé. La vidéo est en cours de lecture, et quand on clique sur le bouton **Pause**, le bouton se retransforme en bouton **Lecture** pour permettre de relancer la vidéo si besoin.

Dans ce cas, il faut que les images qui figurent sur les boutons soient décrites textuellement : « lecture » ou « pause », selon le cas. Il faut également que le changement d'état (la transformation du bouton de **Lecture** à **Pause**, et inversement) soit bien retranscrite.

ATTENTION wmode transparent = danger

Pour positionner un objet Flash dans une page web en CSS, il faut modifier le paramètre par défaut de l'attribut `wmode` (`window`) si on veut que l'objet soit dans le flux.

Avec un `wmode` opaque, seul JavaScript peut positionner l'élément. Les développeurs utilisent donc de préférence le mode transparent, ce qui permet de mettre de la couleur à travers l'objet, par exemple.

Or, un `wmode` transparent est équivalent à un `display:none` en CSS, ce qui signifie que l'objet ne sera pas restitué par les lecteurs d'écran. C'est donc un mode à proscrire, qui rend l'objet forcément inaccessible. Pour plus de détail sur les techniques Flash et l'accessibilité, nous vous renvoyons vers le site de WebAIM (en anglais) :

► <http://ur1.ca/9zr50a>

a. <http://webaim.org/techniques/flash/>

Le mieux pour évaluer cela est encore d'utiliser un lecteur d'écran comme NVDA ou Orca pour vérifier que tout est bien rendu par le lecteur d'écran. Vous pouvez également consulter la procédure de test pour découvrir d'autres outils dans le document méthodologie de test sur le site AccessiWeb.

Médias de décoration et autres cas particuliers

Cette section pourrait se résumer en un mot : servez-vous de votre tête ! L'accessibilité sert à transmettre l'information à tous sans discriminer les personnes en situation de handicap. Mais évidemment, si un média consiste en une jolie animation faisant tomber de la neige sur votre page d'accueil pour les fêtes de fin d'année, inutile de sous-titrer « la neige tombe » ou d'audiodécrire « ploc ploc » pour figurer les flocons de neige.

Comme pour les images, il s'agit d'abord d'identifier si le média véhicule une information. Si ce n'est pas le cas, la plupart des critères de la thématique 4 sont non applicables.

De même si le texte environnant apporte la même information. On peut alors considérer que le média est lui-même une alternative au texte. Notons toutefois que, dans ce cas précis, AccessiWeb fait un cas particulier, y compris pour le critère 4.3 sur le sous-titrage. Or, même si une vidéo est effectivement conçue comme une alternative au texte, la perception et l'expérience des utilisateurs sont différentes d'une version texte. Faire du sous-titrage reste donc, de notre point de vue, essentiel, même si AccessiWeb autorise une dérogation.

DÉBAT Accessibilité et droit d'auteur

AccessiWeb 2.2 propose une gestion de cas particuliers également pour des raisons de droit d'auteur. Ainsi, si « le média temporel est protégé par un copyright, un droit d'auteur ou une clause juridique particulière qui en interdit la modification ou la création de copie modifiée » (<http://ur1.ca/9z39t>^a), vous pouvez considérer les critères 4.1, 4.2, 4.3, 4.5, 4.9, 4.11, 4.13 et 4.16 comme non applicables.

Dit autrement, à moins d'être l'auteur de l'œuvre que vous vous apprêtez à diffuser, ou d'avoir une autorisation expresse de l'auteur, vous n'êtes plus dans l'obligation de faire une transcription, de proposer du sous-titrage ou de l'audiodescription, ou encore de décrire le média statique qui peut très bien être une image porteuse d'information réalisée en Flash. Ces procédés sont en effet considérés comme une modification de l'œuvre.

Mais que viennent faire ces considérations juridiques dans un référentiel technique qui se doit en principe de vérifier si oui ou non la page web est accessible ?

Par ailleurs, sommes-nous sûrs qu'il est légalement répréhensible de sous-titrer un média sur le Web, alors que des lois imposent au contraire de rendre accessibles tous ses contenus web ?

Ces questions ont fait l'objet d'intenses échanges sur le forum dédié à la mise à jour du référentiel AccessiWeb et sur la liste de discussion du groupe de travail AccessiWeb. La décision de BrailleNet a malheureusement été de maintenir ces cas particuliers. Si vous êtes dans un processus de labellisation, c'est donc sans doute une bonne nouvelle pour vous. Bien que vous puissiez malgré tout considérer, en ce qui vous concerne, ces critères comme applicables.

En revanche, si votre préoccupation première est de rendre vos contenus accessibles, oubliez tout simplement ce cas particulier...

Pour un aperçu du débat, vous pouvez consulter le billet de blog suivant :

► <http://ur1.ca/9z3f6>^b

a. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#cpCrit4-

b. <http://www.armonyaltinier.fr/index.php?post/2011/09/16/L-accessibilite%20du-Web-menace-par-le-droit-d-auteur>

Tableaux (thématique AccessiWeb 5)

« Donner à chaque tableau de données, un résumé et un titre pertinent, identifier clairement les cellules d'en-tête, utiliser un mécanisme pertinent pour lier les cellules de données aux cellules d'en-tête. Pour chaque tableau de mise en forme, veiller à sa bonne linéarisation. »

<http://ur1.ca/abzgg>¹

Un tableau sert avant tout à présenter des données. Or, surtout quand le tableau est complexe avec des données croisées en colonne et en ligne (mais même sur un tableau simple), la lecture nécessite qu'on identifie bien quels sont les titres de colonnes ou de lignes pour savoir comment comprendre les données.

En général, cette identification s'effectue visuellement par une mise en gras des titres de colonnes et/ou de lignes. Mais comment restituer l'information pour les utilisateurs aveugles dotés d'un lecteur d'écran ?

Par ailleurs, les tableaux servent aussi parfois à réaliser des mises en page. Ce n'est pas idéal, et des solutions bien plus souples et maintenables existent en CSS, mais c'est possible. Comment ce genre de tableau va-t-il être interprété par les technologies d'assistance ?

INFO Critères de la thématique « Tableaux »

La thématique « Tableaux » contient huit critères, tous de niveau Bronze.
Nous traiterons ici de tous les critères présents dans la thématique.

1. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#tableaux

Tableaux de données et tableaux de présentation

AccessiWeb distingue deux types de tableaux : les tableaux de données et les tableaux de mise en forme.

IDÉE REÇUE « Mon site est accessible, il n'utilise pas de tableaux ! »

On peut utiliser des tableaux HTML et faire des sites accessibles. Il est d'ailleurs recommandé d'utiliser des tableaux HTML quand il s'agit de présenter des données. Et même si la bonne pratique privilégie l'utilisation de feuilles de styles CSS pour la mise en page, il est possible, en termes d'accessibilité, de réaliser une mise en page en tableaux HTML à condition que la linéarisation reste compréhensible. Ceci dit, si vous pouvez éviter, c'est quand même préférable...

Les tableaux de présentation

Les tableaux de mise en forme ne servent pas à présenter des données, mais juste à disposer les éléments de la page selon un ordre graphique précis.

Pour rendre ce genre de tableaux accessible, il faut veiller à deux choses.

- Il convient de faire en sorte que le tableau, qui va être lu ligne après ligne, c'est-à-dire de haut en bas et de gauche à droite selon l'ordre de lecture français, présente les éléments de façon cohérente. L'information doit pouvoir être comprise quand le tableau est « linéarisé » (critère 5.3). Si vous ne faites pas attention à sa bonne linéarisation, il sera tout simplement inaccessible car les informations seront délivrées dans le désordre, comme sur les figures 8-1 et 8-2 ci-après.
- Il faut faire en sorte que le tableau ne soit pas identifié comme un tableau de données par le lecteur d'écran, mais bien comme un tableau de mise en page (critère 5.8). Pour ce faire, il suffit de ne pas utiliser d'éléments servant à identifier un tableau de données. Les balises à proscrire sont `<caption>`, `<th>`, `<thead>` et `<tfoot>`. Les attributs `scope`, `headers`, `colgroup` et `axis` sont également à bannir.

Qu'aurait-il donc fallu faire dans le cas de la page de la documentation Ubuntu citée en exemple ? Le mieux aurait été d'utiliser les feuilles de styles CSS pour réaliser la mise en page, car la linéarisation n'est tout simplement pas possible autrement. Par ailleurs, des éléments propres aux tableaux de données sont également utilisés, ce qui invalide également le critère 5.8.

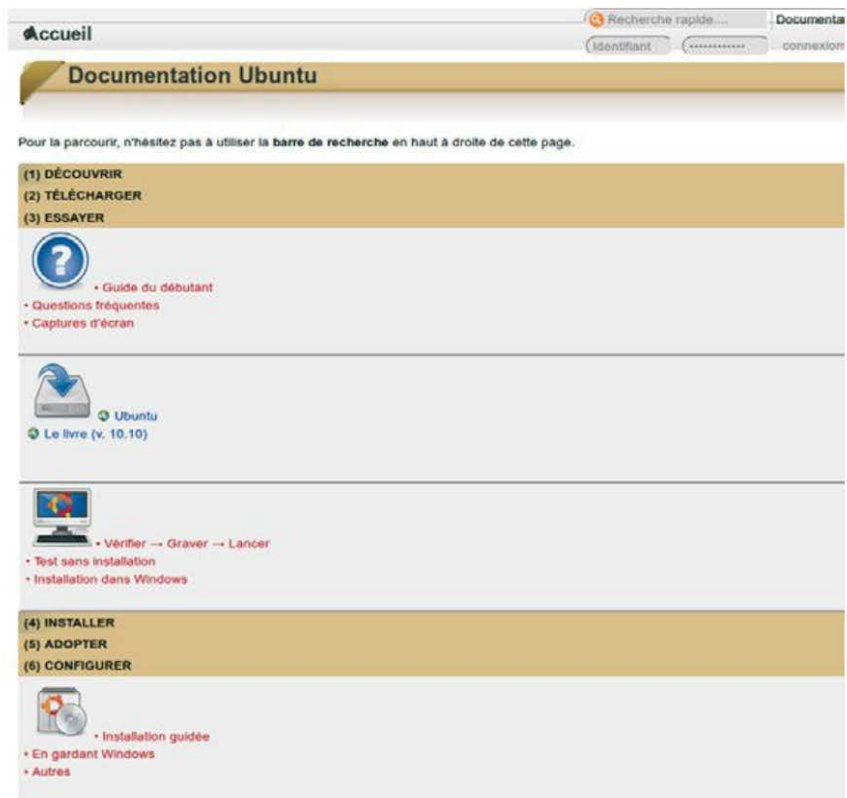
Nous avons vu ce qu'il en était des tableaux de présentation. Qu'en est-il des tableaux de données ?

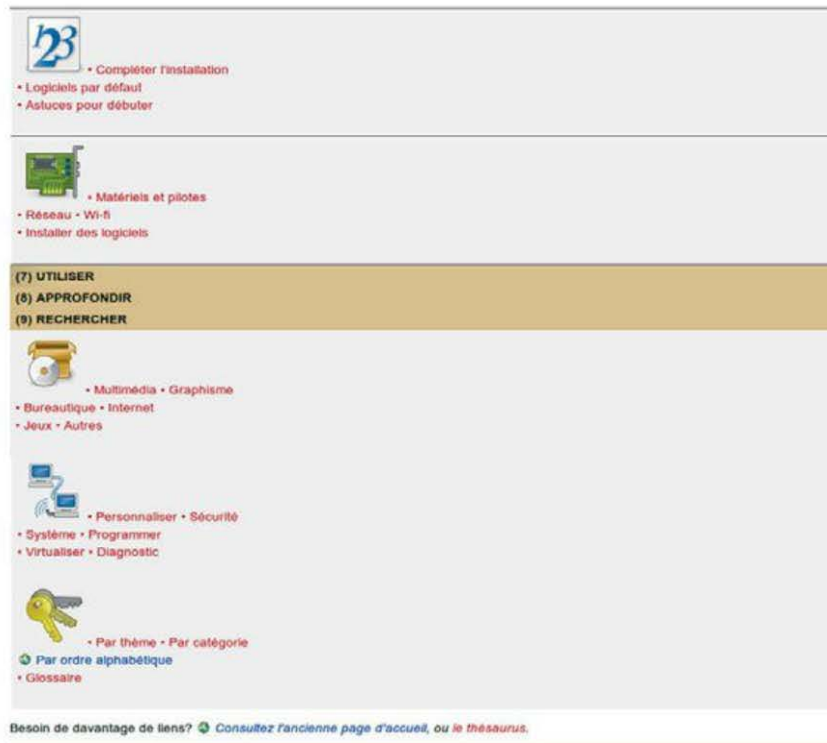
Figure 8–1

Le site de la documentation d'Ubuntu-fr présente ses données à l'aide d'un tableau de mise en forme.
(<http://ubuntu-fr.org>)

**Figure 8–2**

Quand on linéarise la page, on constate que l'information n'est plus compréhensible.
(Suite de la capture d'écran page suivante.)





OUTILS Linéariser la page

Pour avoir un aperçu de la page quand elle est linéarisée, utilisez la barre d'outils de Web Developer. Dans le menu *Divers*, choisissez *Linéariser la page*.

Les tableaux de données

Le principal objectif pour rendre un tableau de données accessible va être de permettre aux lecteurs d'écran d'identifier quelles sont les en-têtes de colonnes ou de lignes (balise `<th>`) et de relier les données du tableau (données contenues dans la balise `<td>`) à leur en-tête. Par ailleurs, pour bien identifier à quoi correspond le tableau, il convient de lui ajouter un titre (balise `<caption>`).

Et comme toujours, tout ceci devra être fait de manière pertinente...

Nous allons voir en détail dans la section suivante comment réaliser un tableau de données accessible.

À NOTER L'attribut `summary` obsolète en HTML5

Le critère 5.1 d'AccessiWeb 2.2 demande à ce qu'un attribut `summary` soit présent sur la balise `<table>` des tableaux de données. Le critère 5.2 précise, quant à lui, que le contenu de l'attribut `summary` doit être pertinent.

Or, cet attribut est obsolète en HTML5, et il est en pratique assez difficile de décrire un `summary` différent du titre du tableau contenu dans la balise `<caption>`.

Si vous souhaitez vous faire labelliser, il vous faudra donc, malgré tout, mettre un attribut `summary` et, en conséquence, coder vos pages en HTML4 pour ne pas invalider le code source (critère 8.2). Mais si l'accessibilité est votre seul souci, vous pouvez ignorer les critères 5.1 et 5.2.

Notons quand même qu'il est autorisé par AccessiWeb, même si non obligatoire, d'utiliser un attribut `summary` sur un tableau de présentation. Mais dans ce cas, pour qu'il soit pertinent, son contenu doit être vide pour valider le critère 5.2, de même qu'un `alt` vide pour les images.

Réaliser un tableau de données accessible

Nous allons ici vous présenter quelques exemples, mais s'ils s'avéraient insuffisants dans votre cas particulier, nous vous encourageons à lire la documentation officielle (en anglais) sur le site du W3C, notamment la section consacrée au rendu des tableaux sur des agents utilisateurs non visuels, intitulée « *Table rendering by non-visual user agents* » : <http://ur1.ca/9zcut²>.

Exemple d'un tableau simple

Commençons par étudier un tableau de données simple, avec des en-têtes de colonnes uniquement sur la première ligne (figure 8-3, page suivante).

2. <http://www.w3.org/TR/html4/struct/tables.html#h-11.4>

Figure 8-3
Part des femmes candidates et
élues à l'Assemblée nationale
en 2012 sur le site de l'INSEE
(<http://ur1.ca/9zcxz>^a)

■ **Part des femmes candidates et élues à l'Assemblée nationale en 2012**

Part des femmes candidates et élues à l'Assemblée nationale en 2012		
en %		
Années	Part des femmes parmi les candidats	Part des femmes parmi les élus
1958	2,3	1,3
1962	2,4	1,7
1967	2,9	1,9
1968	3,3	1,7
1973	6,6	1,7
1978	16,3	4,0
1981	13,1	5,5
1986	25,1	5,8
1988	11,9	5,6
1993	19,5	5,9
1997	23,2	10,8
2002	39,3	12,1
2007	41,6	18,5
2012	40,0	26,9

Champ : France.

Source : ministère de l'Intérieur.

a. http://www.insee.fr/fr/themes/tableau.asp?reg_id=0&ref_id=NATSOS05506

Voici le code source de ce tableau :

```
<table id="NATSOS05506" summary="Part des femmes candidates et élues à  
l'Assemblée nationale en 2012">  
<caption>Part des femmes candidates et élues à l'Assemblée nationale en 2012</  
caption>  
<thead>  
<tr><td class="unite" colspan="3">en %</td></tr>  
<tr>  
<th scope="col" abbr="Ann">Années</th>  
<th scope="col" abbr="Par">Part des femmes parmi les candidats</th>  
<th scope="col" abbr="Par">Part des femmes parmi les élus</th>  
</tr>  
</thead>  
<tfoot>  
<tr><td class="note" colspan="3">Champ : France.</td></tr>  
<tr><td class="source" colspan="3">Source : ministère de l'Intérieur.</td></tr>  
</tfoot>  
<tbody>  
<tr>  
<td class="etendue-ligne" scope="row" abbr="195">1958</td>  
<td class="tab-chiffre">2,3</td>  
<td class="tab-chiffre">1,3</td>  
</tr>
```



```

<tr>
<td class="etendue-ligne" scope="row" abbr="196">1962</td>
<td class="tab-chiffre">2,4</td>
<td class="tab-chiffre">1,7</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="196">1967</td>
<td class="tab-chiffre">2,9</td>
<td class="tab-chiffre">1,9</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="196">1968</td>
<td class="tab-chiffre">3,3</td>
<td class="tab-chiffre">1,7</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="197">1973</td>
<td class="tab-chiffre">6,6</td>
<td class="tab-chiffre">1,7</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="197">1978</td>
<td class="tab-chiffre">16,3</td>
<td class="tab-chiffre">4,0</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="198">1981</td>
<td class="tab-chiffre">13,1</td>
<td class="tab-chiffre">5,5</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="198">1986</td>
<td class="tab-chiffre">25,1</td>
<td class="tab-chiffre">5,8</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="198">1988</td>
<td class="tab-chiffre">11,9</td>
<td class="tab-chiffre">5,6</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="199">1993</td>
<td class="tab-chiffre">19,5</td>
<td class="tab-chiffre">5,9</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="199">1997</td>
<td class="tab-chiffre">23,2</td>
<td class="tab-chiffre">10,8</td>
</tr>

```

```
<tr>
<td class="etendue-ligne" scope="row" abbr="200">2002</td>
<td class="tab-chiffre">39,3</td>
<td class="tab-chiffre">12,1</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="200">2007</td>
<td class="tab-chiffre">41,6</td>
<td class="tab-chiffre">18,5</td>
</tr>
<tr>
<td class="etendue-ligne" scope="row" abbr="201">2012</td>
<td class="tab-chiffre">40,0</td>
<td class="tab-chiffre">26,9</td>
</tr>
</tbody>
</table>
```

Le tableau contient bien un titre (balise `<caption>`) et même un résumé (attribut `summary` sur la balise `<table>`).

Notons toutefois que le titre et le résumé du tableau sont identiques. Or, pour valider le critère de pertinence du résumé, il doit être différent du titre.

On voit par ailleurs qu'il y a bien des cellules d'en-têtes déclarées (`<th>`). Ce tableau étant bien fait, on observe que les cellules de données ont été reliées à leur en-tête. Pour relier en-têtes et données dans ce cas précis, le plus simple a été d'utiliser l'attribut `scope` sur les balises `<th>`. Le terme anglais *scope* peut se traduire en français par la portée, l'étendue. L'expression *scope of action* désigne, par exemple, un rayon d'action.

Et c'est justement ce à quoi sert cet attribut. Il va indiquer aux technologies d'assistance, notamment le lecteur d'écran, quelle est la portée, le rayon d'action de l'en-tête. L'en-tête porte-t-il sur la colonne ou sur la ligne ?

- S'il s'agit d'un en-tête de colonne, on associe la valeur `col` à l'attribut `scope` :
`<th scope="col">Titre de la colonne</th>`.
- S'il s'agit d'un en-tête de ligne, on associe la valeur `row` à l'attribut `scope` :
`<th scope="row">Titre de la ligne</th>`.

Notons que les valeurs `col` et `row` portent respectivement sur toute la colonne et sur toute la ligne. L'utilisation de l'attribut `scope` se limite donc aux tableaux dont les en-têtes visent la colonne ou la ligne entière.

À NOTER Un attribut scope sur des balises <td>

Même si le tableau n'a pas d'en-tête de ligne, chaque ligne s'est vu affecter un attribut `scope`. L'attribut `scope` fait que la cellule de données se comporte comme une cellule d'en-tête. Cela permet aux synthèses vocales de fournir l'année correspondante en cas de demande de l'utilisateur ou de la mentionner immédiatement avant l'énoncé du contenu de chaque cellule, selon le paramétrage du lecteur d'écran.

Enfin, vous remarquerez qu'un attribut `abbr` a été utilisé sur les balises `<th>`. Vous connaissez peut-être déjà la balise `<abbr>` servant à spécifier une abréviation (voir la thématique 9 « Structuration de l'information », dans le chapitre 10). Or, dans un tableau, c'est un attribut `abbr` que l'on place sur une balise `<th>` pour rendre la lecture du tableau par un lecteur d'écran plus fluide. L'attribut est placé dans le cadre d'un intitulé assez long et résume l'intitulé de la ligne ou de la colonne.

Malheureusement, l'utilisation est ici erronée et ne permet pas de comprendre le sens de ce que contient l'en-tête. On voit en effet « Ann » pour année (dans `<th>` qui ne nécessitait pas d'attribut `abbr`), et le même « Par » pour la part des femmes candidates et la part des femmes élues, ce qui ne permet pas de connaître le sens de l'en-tête.

Dans le cas présent, une bonne solution aurait effectivement été d'ajouter des attributs `abbr`, mais en les renseignant comme suit :

```
<!-- On supprime l'attribut abbr sur la première colonne -->
<th scope="col">Années</th>

<!-- On modifie le contenu de l'attribut abbr des deux autres colonnes pour
les distinguer -->
<th scope="col" abbr="candidates">Part des femmes parmi les candidats</th>
<th scope="col" abbr="élues">Part des femmes parmi les élus</th>
```

Enfin, tous les attributs `abbr` ajoutés sur les années sont non seulement erronés, mais également inutiles. Le mieux aurait été de les supprimer.

Nous venons de voir un exemple assez simple. On constate que malgré les efforts prometteurs de l'INSEE pour bien faire, il reste quelques erreurs gênantes pour l'accessibilité, en raison de l'utilisation erronée de l'attribut `abbr`.

Il est donc essentiel de bien comprendre le rôle de chaque règle pour éviter une application systématique et abusive risquant de créer des problèmes d'accessibilité.

Exemple d'un tableau simple à double entrées

Prenons maintenant un exemple un peu plus complexe, avec un tableau à double entrées, contenant à la fois des en-têtes de lignes et de colonnes (figure 8-4, page suivante).

Figure 8–4
Handicaps et discriminations
chez les personnes de 25
à 54 ans, tableau publié
sur le site de l’INSEE

Handicaps et discriminations chez les personnes de 25 à 54 ans						
	Population selon les handicaps					
	Ensemble	Aucun handicap	Au moins un handicap	dont : au moins un handicap d'ordre		
				cognitif	moteur	sensoriel
Répartition de la population	100	90,4	9,6	4,6	4,1	3,0
Personnes ayant déclaré avoir subi des discriminations liées à la santé ou au handicap	6	4	23	31	26	19
Parmi elles, personnes ayant déclaré, pour des raisons liées à un problème de santé ou de handicap :						
avoir subi des insultes, des moqueries	68	71	65	67	62	60
avoir été mises à l'écart des autres	41	38	47	49	42	54
avoir été traitées injustement	33	27	42	44	44	42
s'être vu refuser un droit	17	14	22	21	26	23
En raison :						
de l'apparence	57	65	44	39	51	43
de préjugés, de jugements de valeur	30	19	47	50	46	50
de limitations liées au handicap	23	14	38	40	41	40
d'attitudes inhabituelles pour les autres	11	6	18	23	16	15

Champ : France, personnes de 25 à 54 ans vivant en ménage ordinaire, soit 25,3 millions de personnes.

Source : Insee, enquête Handicap santé 2008 - volet ménages.

Ce tableau pourrait en fait être décomposé en trois tableaux. Pour cet exemple, nous ne nous occuperons que de la première partie du tableau (figure 8-5).

Figure 8–5
Première partie du tableau
de l’INSEE

Handicaps et discriminations chez les personnes de 25 à 54 ans						
	Population selon les handicaps					
	Ensemble	Aucun handicap	Au moins un handicap	dont : au moins un handicap d'ordre		
				cognitif	moteur	sensoriel
Répartition de la population	100	90,4	9,6	4,6	4,1	3,0
Personnes ayant déclaré avoir subi des discriminations liées à la santé ou au handicap	6	4	23	31	26	19

Voici le code source de ce tableau :

```
<table id="tableau2" summary="Handicaps et discriminations chez les personnes de
25 à 54 ans">
<caption>Handicaps et discriminations chez les personnes de 25 à 54 ans
</caption>
<tfoot>
<tr><td class="note" colspan="7">Champ : France, personnes de 25 à 54 ans
vivant en ménage ordinaire, soit 25,3 millions de personnes.</td></tr>
<tr><td class="source" colspan="7">Source : Insee, enquête Handicap santé
2008 - volet ménages.</td></tr>
</tfoot>
<tbody>
<tr>
<th rowspan="3">
</th>
<th colspan="6">Population selon les handicaps</th>
</tr>
<tr>
<th rowspan="2">Ensemble</th>
<th rowspan="2">Aucun handicap</th>
<th rowspan="2">Au moins un handicap</th>
<th colspan="3">dont : au moins un handicap d'ordre</th>
</tr>
<tr>
<th>cognitif</th>
<th>moteur</th>
<th>sensoriel</th>
</tr>
<tr>
<td colspan="7">
<!-- Cette cellule contient un titre, ce devrait être une balise th -->
<td class="TexteTableau">
<strong>Répartition de la population</strong>
</td>
<td class="TexteTableau">
<strong>100</strong>
</td>
<td class="TexteTableau">90,4</td>
<td class="TexteTableau">9,6</td>
<td class="TexteTableau">4,6</td>
<td class="TexteTableau">4,1</td>
<td class="TexteTableau">3,0</td>
</tr>
<tr>
<td colspan="7">
<!-- Cette cellule contient un titre, ce devrait être une balise th -->
<td class="TexteTableau">
```



```
<strong>Personnes ayant déclaré avoir subi des discriminations liées à la santé
ou au handicap</strong>
</td>
<td class="TexteTableau">
<strong>6</strong>
</td>
<td class="TexteTableau">4</td>
<td class="TexteTableau">23</td>
<td class="TexteTableau">31</td>
<td class="TexteTableau">26</td>
<td class="TexteTableau">19</td>
</tr>

<!-- Nous fermons la balise table pour l'exemple, bien que le tableau d'origine
ne soit pas terminé -->
</tbody>
</table>
```

Le tableau est certes un peu plus complexe, mais nous avons déjà vu, avec l'attribut `scope`, comment le réaliser.

Notons toutefois qu'à part l'utilisation d'un titre de tableau pertinent via la balise `<caption>`, le résumé du tableau reste identique au titre (ce qui invalide le critère 5.2) et les cellules de données ne sont plus reliées aux cellules d'en-têtes.

Il suffit pourtant d'indiquer un attribut `scope="col"` pour les en-têtes de colonnes, et un attribut `scope="row"` pour les en-têtes de lignes, comme précédemment.

Nous allons maintenant découvrir deux nouvelles valeurs pour l'attribut `scope` : `colgroup` et `rowgroup`. La valeur `rowgroup` signifie que l'en-tête de la cellule ayant cette valeur fournit une information de titre pour le reste du groupe de lignes qui dépendent d'elle. Il en va de même pour la valeur `colgroup`, mais appliqué au groupe de colonnes qui dépendent de la cellule d'en-tête affectée de l'attribut `scope="colgroup"`.

En pratique, voilà ce que donnerait le code corrigé :

```
<!-- On modifie le résumé pour qu'il soit différent du titre du tableau. On voit
que c'est difficile de le faire de façon vraiment intelligible. Heureusement que
l'attribut summary est déprécié en HTML5. -->
<table id="tableau2" summary="Handicaps et discrimination chez les personnes de
35 à 54 ans présentés par population selon le handicap, résultat pour
l'ensemble, pour aucun handicap, dont au moins un handicap cognitif, moteur ou
sensoriel">
<caption>Handicaps et discriminations chez les personnes de 25 à 54 ans</
caption>
<tfoot>
<tr><td class="note" colspan="7">Champ&nbsp;: France, personnes de 25 à 54 ans
vivant en ménage ordinaire, soit 25,3 millions de personnes.</td></tr>
```

```

<tr><td class="source" colspan="7">Source&nbsp;: Insee, enquête Handicap santé
2008 - volet ménages.</td></tr>
</tfoot>
<tbody>
<tr>

<!-- Cellule vide -->
<td rowspan="3">
</td>

<!-- On ajoute un attribut scope valant pour l'ensemble des 6 colonnes dont
dépendent la cellule -->
<th colspan="6" scope="colgroup">Population selon les handicaps</th>
</tr>
<tr>

<!-- L'en-tête ici vaut pour la colonne, donc on met un attribut scope
correspondant -->
<th rowspan="2" scope="col">Ensemble</th>
<th rowspan="2" scope="col">Aucun handicap</th>
<th rowspan="2" scope="col">Au moins un handicap</th>

<!-- L'en-tête ici vaut pour les 3 colonnes qui en dépendent -->
<th colspan="3" scope="colgroup">dont&nbsp;: au moins un handicap d'ordre</th>
</tr>
<tr>

<!-- L'en-tête vaut ici pour la colonne -->
<th scope="col">cognitif</th>
<th scope="col">moteur</th>
<th scope="col">sensoriel</th>

<!-- Cette cellule contient un titre, on remplace la balise td par une balise th
-->
<th class="TexteTableau" scope="row">
<strong>Répartition de la population</strong>
</th>
<td class="TexteTableau">
<strong>100</strong>
</td>
<td class="TexteTableau">90,4</td>
<td class="TexteTableau">9,6</td>
<td class="TexteTableau">4,6</td>
<td class="TexteTableau">4,1</td>
<td class="TexteTableau">3,0</td>
</tr>
<tr>

```

```
<!-- Cette cellule contient un titre, on remplace la balise td par une balise th
-->
<th class="TexteTableau" scope="row">
<strong>Personnes ayant déclaré avoir subi des discriminations liées à la santé
ou au handicap</strong>
</th>
<td class="TexteTableau">
<strong>6</strong>
</td>
<td class="TexteTableau">4</td>
<td class="TexteTableau">23</td>
<td class="TexteTableau">31</td>
<td class="TexteTableau">26</td>
<td class="TexteTableau">19</td>
</tr>

<!-- Nous fermons la balise table pour l'exemple, bien que le tableau d'origine
ne soit pas terminé -->
</tbody>
</table>
```

À SAVOIR Comment traiter la case vide ?

Dans le cas d'un tableau à double entrée, il y a forcément une cellule du tableau vide, celle qui se situe à la jonction entre la ligne d'en-tête et la colonne d'en-tête.

L'idéal serait de permettre aux technologies d'assistance d'ignorer cette cellule, mais ce n'est pas possible techniquement. La meilleure solution consiste donc à traiter cette cellule comme une cellule de données (balise `<td>` sans attribut) et de la laisser vide.

Cet exemple était certes un peu plus complexe dans la forme, mais au fond, la mise en accessibilité n'était pas plus compliquée que dans le premier exemple. Nous avons pu utiliser l'attribut `scope` puisque l'étendue de chaque en-tête visait à chaque fois la colonne ou la ligne entière.

Corsons un peu les choses avec l'exemple suivant.

Exemple d'un tableau complexe à en-têtes multiples

Dans cet exemple, nous vous proposons un tableau vraiment complexe, avec des en-têtes non seulement sur les colonnes et les lignes, mais également à l'intérieur du tableau lui-même (figure 8-6).

Figure 8–6

Données économiques
et financières concernant
le secteur réel, tableau
publié sur le site de l'INSEE
(<http://ur1.ca/9zcve>^a)

SECTEUR RÉEL					Pour en savoir plus
Données et composantes selon la norme du SDDS	Unité	Date de la dernière observation	Derniers chiffres	Chiffres précédents	
Comptes nationaux - PIB aux prix de l'année précédente chaînés	milliards d'euros - (cvs-cjo)	T1/12	450,90	450,70	Comptes nationaux trimestriels (Insee)
* Importations	"	"	133,4	132,2	
* Dépenses de consommation des ménages	"	"	373,8	372,7	
* Dépenses de consommation des APU	"	"	112,00	111,40	
* Dépenses de consommation des ISBLSM	"	"	9,3	9,3	
* FBCF	"	"	86,60	87,20	
* Variations de stocks 1/	"	"	-0,8	-0,1	
* Exportations	"	"	125,7	125,5	
Comptes nationaux - PIB aux prix courants	milliards d'euros - (cvs-cjo)	T1/12	504,1	502,7	
* Importations	"	"	150,9	148,4	
* Dépenses de consommation des ménages	"	"	415,6	412,5	
* Dépenses de consommation des APU	"	"	123,7	123,1	
* Dépenses de consommation des ISBLSM	"	"	10,5	10,4	
* FBCF	"	"	102,3	102,3	
* Variations de stocks 1/	"	"	-1,6	-1,4	
* Exportations	"	"	138,7	137,7	
Indice de la production industrielle (hors énergie et IAA) 2/	2005=100 (cvs-cjo)	juin 12	90,1	90,8	
Enquêtes de conjoncture					
* Enquête mensuelle dans l'industrie manufacturière - production passée -	solde d'opinion des industriels en % (cvs)	juillet 12	-9	-7	
* Enquête trimestrielle dans l'industrie manufacturière - taux d'utilisation des capacités de production - (*)	% (cvs)	juillet 12	80	81	
Emploi salarié 4/	en milliers (cvs)	T1/12	18 074,10	18 033,30	
Taux de chômage au sens du BIT	% en fin de mois (cvs)				
* France (y compris DOM)	% (cvs)	T1/12	10,0	9,8	
* France métropolitaine	% (cvs)	T1/12	9,6	9,3	
Salaires - Revenus 3/ 5/					
* Salaire horaire de base des ouvriers (SHBO)	variation trimestrielle, en %	T1/12	1	0,3	
* Salaire mensuel de base (SMB)	variation trimestrielle, en %	T1/12	0,9	0,3	
Indice des prix à la consommation	1998=100	juillet 12	125,8	126,3	
Indices de prix à la production dans l'industrie - Produits manufacturés (y c. IAA - indices hors TVA)	2005=100	juin 12	116,9	118	

(*) Avertissement : rupture de série en janvier 2004 due à la nouvelle formulation de la question relative au taux d'utilisation. Les données sont présentées en données brutes à compter de cette date. Pour plus d'informations, consulter la note méthodologie de l'enquête trimestrielle dans l'industrie.

a. http://www.insee.fr/fr/themes/theme.asp?theme=17&sous_theme=4#FR_1

Comme pour l'exemple précédent, nous ne nous intéresserons qu'à un extrait du tableau pour ne pas avoir un code trop long à étudier. Nous allons ainsi étudier uniquement la deuxième partie du tableau (figure 8-7, page suivante).

Figure 8-7
Deuxième partie du tableau,
à partir des données de
conjoncture

SECTEUR RÉEL					
Données et composantes selon la norme du SDDS	Unité	Les résultats			Pour en savoir plus
		Date de la dernière observation	Derniers chiffres	Chiffres précédents	
Enquêtes de conjoncture					
* Enquête mensuelle dans l'industrie manufacturière - production passée -	solde d'opinion des industriels en % (cvs)	juillet 12	-9	-7	
* Enquête trimestrielle dans l'industrie manufacturière - taux d'utilisation des capacités de production - (*)	% (cvs)	juillet 12	80	81	
Emploi salarié 4/	en milliers (cvs)	T1/12	18 074,10	18 033,30	
Taux de chômage au sens du BIT	% en fin de mois (cvs)				
* France (y compris DOM)	% (cvs)	T1/12	10,0	9,8	
* France métropolitaine	% (cvs)	T1/12	9,6	9,3	
Salaires - Revenus 3/ 5/					
* Salaire horaire de base des ouvriers (SHBO)	variation trimestrielle, en %	T1/12	1	0,3	
* Salaire mensuel de base (SMB)	variation trimestrielle, en %	T1/12	0,9	0,3	
Indice des prix à la consommation	1998=100	juillet 12	125,8	126,3	
Indices de prix à la production dans l'industrie - Produits manufacturés (y c. IAA - indices hors TVA)	2005=100	juin 12	116,9	118	

(*) Avertissement : rupture de série en janvier 2004 due à la nouvelle formulation de la question relative au taux d'utilisation. Les données sont présentées en données brutes à compter de cette date. Pour plus d'informations, consulter la note méthodologie de l'enquête trimestrielle dans l'industrie.

Voici le code source de ce tableau :

```
<table id="FR_1" summary="SECTEUR RÉEL">
<caption>SECTEUR RÉEL</caption>
<tfoot>
<tr><td class="note" colspan="6">(*) Avertissement : rupture de série en
janvier 2004 due à la nouvelle formulation de la question relative au taux
d'utilisation. Les données sont présentées en données brutes à compter de cette
date. Pour plus d'informations, consulter la note méthodologie de l'enquête
trimestrielle dans l'industrie.</td></tr>
</tfoot>
<tbody>
<tr>
<!-- Les en-têtes ne sont reliés à aucune cellule -->
<th rowspan="2">Données et composantes selon la norme du SDDS</th>
<th rowspan="2">Unité</th>
<th colspan="3">Les résultats</th>
<th rowspan="2">Pour en savoir plus</th>
</tr>
<tr>
<th>Date de la dernière observation</th>
<th>Derniers chiffres</th>
<th>Chiffres précédents</th>
</tr>
<tr>
<td>Enquêtes de conjoncture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Enquête mensuelle dans l'industrie
manufacturière - production passée -</td>
<td>solde d'opinion des
industriels en %
(cvs)</td>
<td>juillet 12</td>
<td>-9</td>
<td>-7</td>
<td></td>
</tr>
<tr>
<td>* Enquête trimestrielle dans l'industrie
manufacturière - taux d'utilisation des
capacités de production - (*)</td>
<td>% (cvs)</td>
<td>juillet 12</td>
<td>80</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>Emploi salarié 4/</td>
<td>en milliers (cvs)</td>
<td>T1/12</td>
<td>18 074,10</td>
<td>18 033,30</td>
<td></td>
</tr>
<tr>
<td>Taux de chômage au sens du BIT</td>
<td>% en fin de mois
(cvs)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* France (y compris DOM)</td>
<td>% (cvs)</td>
<td>T1/12</td>
<td>10,0</td>
<td>9,8</td>
<td></td>
</tr>
<tr>
<td>* France métropolitaine</td>
<td>% (cvs)</td>
<td>T1/12</td>
<td>9,6</td>
<td>9,3</td>
<td></td>
</tr>
<tr>
<td>Salaires - Revenus 3/ 5/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Salaire horaire de base des ouvriers
(SHBO)</td>
<td>variation
trimestrielle, en %</td>
<td>T1/12</td>
<td>1</td>
<td>0,3</td>
<td></td>
</tr>
<tr>
<td>* Salaire mensuel de base (SMB)</td>
<td>variation
trimestrielle, en %</td>
<td>T1/12</td>
<td>0,9</td>
<td>0,3</td>
<td></td>
</tr>
<tr>
<td>Indice des prix à la consommation</td>
<td>1998=100</td>
<td>juillet 12</td>
<td>125,8</td>
<td>126,3</td>
<td></td>
</tr>
<tr>
<td>Indices de prix à la production
dans l'industrie - Produits
manufacturés (y c. IAA - indices
hors TVA)</td>
<td>2005=100</td>
<td>juin 12</td>
<td>116,9</td>
<td>118</td>
<td></td>
</tr>
</tbody>
</table>
```


149

150

```

<!-- Cette cellule devrait être une cellule d'en-tête -->
<td class="tab-intertitre">Indices de prix à la production dans l'industrie -
Produits manufacturés (y c. IAA - indices hors TVA)</td>
<td class="tab-intertitre">2005=100</td>
<td class="tab-intertitre">juin 12</td>
<td class="tab-chiffre">116,9</td>
<td class="tab-chiffre">118</td>
</tr>
</tbody>
</table>

```

Pour rendre ce tableau accessible, l'utilisation de l'attribut `scope` ne va plus être possible. En effet, nous avons vu que `scope` permettait d'indiquer le champ d'application de l'en-tête. Or, ce champ d'application repose sur l'ensemble des cellules de la ligne ou de la colonne.

Dans l'exemple que nous avons pris, nous voyons que certaines données relèvent d'une cellule d'en-tête qui se trouve dans la colonne, mais qu'elle ne relève pas de l'en-tête figurant juste au-dessus dans cette même colonne.

Ainsi, la cellule de données « salaire mensuel de base (SMB) » relève de l'en-tête « Salaires - Revenus 3/ 5/ », ainsi que de l'en-tête « Données et composantes selon la norme du SDDS », mais pas de l'en-tête « Enquêtes de conjoncture » qui se trouve dans cette même colonne plus haut. Or, si nous mettions un `scope="col"` sur la cellule d'en-tête « Enquêtes de conjoncture », cela s'appliquerait à toutes les cellules de la colonne figurant au-dessous.

Il va donc falloir relier plus précisément les cellules de données à leurs en-têtes respectifs. Pour ce faire, nous allons commencer par utiliser un identifiant unique par en-tête (attribut `id` avec la valeur de son choix, qui doit être unique). Puis, nous allons relier chaque cellule de données à ses en-têtes grâce à l'attribut `headers` qui reprendra en valeur l'identifiant de la cellule d'en-tête à laquelle elle est reliée.

Reprenons notre exemple, et voyons comment corriger le code :

```

<!-- On met un résumé différent du titre -->
<table id="FR_1" summary="Données économiques et financières SECTEUR RÉEL
présentées en six colonnes, avec 1/les données et composantes selon la norme du
SDDS, 2/1'unité, 3/les résultats classés par date de la dernière observation, 4/
les derniers chiffres et 5/les chiffres précédents, et enfin 6/une colonne pour
en savoir plus">
<caption>SECTEUR RÉEL</caption>
<tfoot>

```

152

```

</tr>
<tr>
<th class="tab-intertitre" id="emploi">Emploi salarié 4/</th>
<td headers="unit emploi" class="tab-intertitre">en milliers (cvs)</td>
<td headers="results obs emploi" class="tab-intertitre">T1/12</td>
<td headers="results last emploi" class="tab-chiffre">18 074,10</td>
<td headers="results before emploi" class="tab-chiffre">18 033,30</td>
</tr>
<tr>
<td headers="sdds">Taux de chômage au sens du BIT </td>
<td headers="unit" class="tab-intertitre">% en fin de mois (cvs)</td>
<td class="tab-intertitre">
</td>
<td class="tab-intertitre">
</td>
<td class="tab-intertitre">
</td>
</tr>
<tr>
<td headers="sdds emploi">* France (y compris DOM)</td>
<td headers="unit emploi">% (cvs)</td>
<td headers="results obs emploi">T1/12</td>
<td headers="results last emploi" class="tab-chiffre">10,0</td>
<td headers="results before emploi" class="tab-chiffre">9,8</td>
</tr>
<tr>
<td headers="sdds emploi">* France métropolitaine</td>
<td headers="unit emploi">% (cvs)</td>
<td headers="results obs emploi">T1/12</td>
<td headers="results last emploi" class="tab-chiffre">9,6</td>
<td headers="results before emploi" class="tab-chiffre">9,3</td>
</tr>
<tr>
<th class="tab-intertitre" id="revenus">Salaires - Revenus&nbsp;  3/ 5/</th>
<td class="tab-intertitre">
</td>
<td class="tab-intertitre">
</td>
<td class="tab-intertitre">
</td>
<td class="tab-intertitre">
</td>
</tr>
<tr>
<td headers="sdds revenus">*&nbsp; &nbsp; Salaire horaire de base des ouvriers
(SHB0)</td>
<td headers="unit revenus">variation trimestrielle, en %</td>
<td headers="results obs revenus">T1/12</td>
<td headers="results last revenus" class="tab-chiffre">1</td>

```


Nous voyons que certaines cellules de données sont reliées à pas moins de trois cellules d'en-têtes. Sans l'ajout des attributs `headers` et `id`, il est impossible de comprendre ce tableau dont la complexité rend la lecture linéaire inutilisable. Notons que l'ordre dans lequel sont implémentés les attributs `id` correspond à l'ordre de la restitution par le lecteur d'écran.

Nous avons vu comment renseigner de façon pertinente le résumé (attribut `summary`) et le titre de tableau (balise `<caption>`), à quoi servait l'attribut `abbr` sur une balise `<th>`, et comment utiliser les attributs `scope` et `headers` pour relier les cellules d'en-têtes dans des cas de tableaux simples et complexes.

154

À NOTER Définir des catégories avec l'attribut axis

Il existe un attribut permettant de faire des regroupements conceptuels pour faciliter la recherche des utilisateurs de lecteurs d'écran selon des catégories préétablies. C'est l'attribut `axis`.

Cet attribut étant déprécié en HTML5, nous ne nous attarderons pas dessus. Voici toutefois, pour les curieux, le lien vers la section dédiée sur le site du W3C (en anglais) :

► <http://ur1.ca/9zdpv^a>

a. <http://www.w3.org/TR/html4/struct/tables.html#multi-dimension>

Un tableau accessible, facile ?

Nous avons vu que l'accessibilité d'un tableau de données reposait sur des techniques très simples à mettre en œuvre... à condition de savoir coder. Or, la personne qui va mettre la page du site à jour n'est pas forcément un développeur et se servira du logiciel qui est utilisé pour entrer du contenu.

Deux solutions sont alors possibles :

- le logiciel (CMS) permet de créer des tableaux accessibles, même lorsqu'ils sont complexes : il suffit alors de sensibiliser vos contributeurs à l'accessibilité et de les former à utiliser le CMS à bon escient ;
- le logiciel ne permet pas de créer des tableaux accessibles : vous pouvez alors proposer à vos contributeurs des modèles de tableaux prédéfinis, qu'ils pourront copier-coller selon les besoins et modifier en remplaçant les données de démonstration par les leurs. L'utilisation de ces modèles devra aussi faire l'objet d'une formation, mais cela peut simplifier la tâche des contributeurs tout en conservant un bon niveau d'accessibilité.

Liens et scripts (thématiques AccessiWeb 6 et 7)

Les thématiques AccessiWeb numéro 6 à 7 concernent le traitement des liens (6) et des scripts (7).

Liens

« Donner des intitulés de lien explicites, grâce à des informations de contexte notamment, et utiliser le titre de lien le moins possible. Doubler avec des liens ou un formulaire de navigation les zones réactives d'une image cliquable côté serveur. »

<http://ur1.ca/abv69>¹

Le mot même de *Web*, en anglais, signifie « Toile », pour figurer cette grande toile tissée de liens hypertextes menant d'une page web à une autre. Il est donc plus qu'essentiel que les liens soient accessibles à tous, puisque c'est le fondement même du Web.

Or, qu'est-ce qu'un lien accessible ? C'est un lien qui est identifiable comme tel, qui permet à l'internaute de savoir vers quelle destination il l'emmène, et qui n'entrave pas la navigation en fournissant des éléments inutiles qui seraient restitués par les lecteurs d'écran.

Un lien accessible bénéficiera donc aux personnes aveugles, utilisatrices de lecteurs d'écran. Mais cela est également utile aux personnes ayant un handicap moteur, en leur permettant de sauter les liens qui ne les intéressent pas, leur évitant ainsi une manipulation inutile (taper pour entrer dans le lien, se rendre compte que le contenu n'est pas intéressant, refaire une manipulation pour revenir en arrière...). Enfin, un

1. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#liens

lien accessible bénéficiera aussi aux personnes ayant certains handicaps cognitifs et qui peuvent vite être désorientées dans leur navigation si elles cliquent sur des liens qui ne correspondent pas à ce qu'elles pensaient.

Voyons cela en détail.

INFO Les critères de la thématique « Liens »

La thématique « Liens » contient six critères, dont cinq de niveau Bronze et un de niveau Or. Nous n'aborderons pas le critère 6.5 concernant les images réactives côté serveur, cette technique étant rarement employée. La lecture du référentiel à ce sujet suffira à vous guider si nécessaire. Voir le critère 6.5 en ligne sur le référentiel AccessiWeb 2.2 (liste déployée) :

► <http://ur1.ca/9ylkw>^a

a. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-6-5

Intitulé et titre de lien

Une fois de plus, faisons un petit point de vocabulaire en préambule. AccessiWeb parle « d'intitulé de lien » et de « titre de lien ».

En clair, l'intitulé de lien est le texte que tout le monde pourra lire sur la page, et qui est contenu entre les balises `<a>` et `</a>`. Cela comprend également le contenu de l'attribut `alt` quand il s'agit d'un lien sur une image.

Le titre de lien, quant à lui, concerne l'infobulle qui apparaît au survol et est contenu au sein de la balise `<a>` dans l'attribut `title`.

Un intitulé présent et explicite

Rien de surprenant, l'intitulé de lien doit être présent (critère 6.6) et explicite (critère 6.1). La pertinence de l'intitulé est ici exprimée par le mot « explicite ». Mais cela revient au même. Il s'agit de vérifier si, en lisant le lien, *on en comprend la fonction et la destination*.

CAS PARTICULIER Les liens non explicites à dessein

Notons que si le lien n'est explicite pour personne, les personnes handicapées ne sont pas désavantagées, il ne faut donc pas considérer cela comme une erreur.

Par exemple, si dans un jeu en ligne, trois liens indiquent « Porte 1 », « Porte 2 » et « Porte 3 », ce n'est certes pas explicite, mais c'est fait exprès. Le but étant de conserver le suspense pour les joueurs, il serait contre-productif de considérer ces liens non-explicites comme inaccessibles...

L'ambiguïté du terme « explicite »

Nous avons vu que l'accessibilité web était une matière à part entière, avec son jargon, ses règles, ses référentiels... Et bien qu'il y ait des ponts avec l'ergonomie, ce sont deux matières complémentaires mais différentes. Dans le cas des liens explicites, nous touchons à une ambiguïté qui peut, si elle est mal comprise, engendrer des évaluations ergonomiques plutôt que d'accessibilité.

Prenons l'exemple d'un site de vêtements. Il y a plusieurs liens dont voici les intitulés : « chapeaux », « chaussures », « pantalons »... Ces intitulés sont-ils explicites ? On ne sait pas, en cliquant dessus, si on trouve un choix d'articles que l'on peut acheter, un catalogue, des informations sur les chapeaux... Pourtant, le manque de précision n'est pas un frein en accessibilité, puisque tout le monde nage dans la même imprécision. Il n'y a pas désavantage, donc on peut valider les critères concernés.

Jusqu'ici, tout semble normal. Pourtant, imaginons qu'une erreur fasse qu'en cliquant sur « chapeaux », on trouve une liste de pantalons. Nous serions obligés d'invalider le critère, puisque la destination du lien n'est pas cohérente avec l'intitulé.

« Mais, mais, mais... » me direz-vous, « il n'y a pas de désavantage, puisque tout le monde sera perturbé, non ? Est-ce vraiment un problème d'accessibilité ? » Très bonne question ! Oui, il y a malgré tout un désavantage, puisqu'une personne avec un handicap cognitif peut être totalement désorientée, plus qu'un utilisateur sans handicap.

Tout le problème est donc de permettre une évaluation du critère 6.1 sans chercher à remplacer l'ergonome ou l'expert en référencement qui vont, de leur côté, donner des conseils pour optimiser l'intitulé des liens.

Le seul souci d'un audit en accessibilité est de rechercher, à travers une évaluation du respect des règles d'accessibilité, si personne n'est désavantagé en raison d'un handicap – et de voir si les contenus sont effectivement accessibles. Soyez donc vigilant, si l'on vous demande une évaluation en accessibilité, de ne pas aller au-delà de votre rôle, ce qui n'est pas toujours facile...

Contexte de lien

Une notion importante est introduite pour juger si le lien est explicite. Tandis que le critère 6.3, de niveau Or, exige un intitulé de lien explicite *hors contexte*, le critère 6.1, de niveau Bronze, demande à ce que le lien soit explicite en prenant en compte son contexte. Ce n'est donc pas l'intitulé seul qui est évalué, mais l'intitulé pris dans son contexte.

Le contexte du lien n'est pas à évaluer en fonction du texte environnant tel que perçu visuellement, mais uniquement d'un point de vue programmatique, ce qui rend sa compréhension impossible si on ne connaît pas le HTML (c'est le cas de nombre de contributeurs, pourtant très souvent amenés à créer des liens).

À NOTER L'attribut `title`, un contexte suffisant ?

Depuis la version 2.2 d'AccessiWeb, un attribut `title` venant expliciter l'intitulé du lien permet de valider le critère 6.3 de niveau Or. Il n'est donc plus jamais exigé de liens explicites hors contexte. Notons pourtant que la mauvaise implémentation de cet attribut fait que nombre de personnes aveugles préfèrent nettement un lien explicite hors contexte.

Voyons la définition donnée dans le glossaire AccessiWeb pour « contexte de lien » (<http://ur1.ca/9zpit2>) :

« Le contexte du lien représente les informations supplémentaires (on parle d'informations de contexte) qui peuvent être mises en relation par un programme informatique avec l'intitulé du lien. Les informations de contexte qui permettent de rendre un lien explicite sont les suivantes :

Le contenu du paragraphe (balise `p`) dans lequel le lien texte est présent

Le contenu de l'item de liste (balise `li`) dans lequel le lien texte est présent

Le contenu du titre (balise `H`) précédant le lien texte

Le contenu de la cellule d'en-tête de tableau (balise `th`) associée à la cellule de donnée (balise `td`) dans laquelle le lien texte est présent

Le contenu de la cellule de donnée (balise `td`) dans laquelle le lien texte est présent

Le contenu du titre de lien (attribut `title`) »

Cette définition, très claire, révèle bien que le contexte du lien est à apprécier en fonction de la position dans laquelle est placé le texte contenu entre les balises `<a>` et `</a>`. Si ce texte n'est pas clair, mais que l'information autour, se trouvant entre les balises `<p>` et `</p>` par exemple, le rend explicite, alors le lien sera considéré comme explicite en contexte – en tout cas, au niveau Bronze.

Les liens « cliquez ici », « lire la suite », « en savoir plus »...

Le cas typique de liens non explicites hors contexte sont les liens tels que « cliquez ici », « lire la suite » ou encore « en savoir plus ». Ce type de liens est très fréquent.

Sous AccessiWeb 1.0, ces liens auraient été invalidés systématiquement, dès le niveau Bronze. Avec les WCAG 2 et la notion de « contexte de lien », il devient plus rare qu'un lien devienne non explicite quand il est pris dans son contexte. Les cas d'invalidation du critère 6.1, de niveau Bronze, est donc peu fréquent.

Pourquoi ces mêmes liens qui étaient, hier encore, considérés comme non accessibles sont-ils autorisés aujourd'hui ?

2. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#mContexteLien

Il y a au moins deux raisons à cela.

- Les liens sont un des fondements du Web. À ce titre, toute personne qui publie des contenus est amenée à créer des liens. Et avec la facilité technique accordée à tout le monde pour publier du contenu, il est illusoire de croire qu'on pourra éviter la création de liens non explicites hors contexte. Ce constat d'échec est également associé au constat d'adaptation des personnes handicapées, qui ont appris au fil du temps à contourner ces difficultés en remontant le texte précédant le lien pour savoir à quoi cela renvoie – en se servant justement du contexte, donc.
- Certains lecteurs d'écran sont dotés d'une fonction pour récupérer le contexte programmatique, en lisant le titre précédent, par exemple. Cette évolution technologique rendrait donc accessibles les liens en contexte.

On note toutefois que la nécessité d'un lien explicite hors contexte reste une exigence au niveau Or, niveau d'amélioration.

Cette baisse du niveau d'exigence est assez contestée dans le milieu des experts en accessibilité et parmi les utilisateurs handicapés eux-mêmes. Car même si des astuces permettent de récupérer l'information, cela nécessite des étapes supplémentaires, ce qui ralentit la navigation et la rend plus complexe. Cela rend également caduques certaines fonctionnalités des lecteurs d'écran visant à aider les lecteurs non-voyants à prendre connaissance d'une page web plus rapidement, telle la fonction permettant d'afficher la liste des liens d'une page web.

Notons que l'attribut `title` fait partie intégrante de l'élément `a` et peut donc être restitué si la technologie d'assistance est bien paramétrée.

À SAVOIR Afficher la liste des liens avec un lecteur d'écran

Certains lecteurs d'écran comme NVDA disposent d'une fonctionnalité permettant l'ouverture d'une fenêtre contenant la liste des titres et la liste des liens de la page web.

Ainsi, que ce soit pour découvrir la composition de la page ou aller directement sur une partie de la page qu'on connaît déjà, cette fonctionnalité est très utilisée par les internautes non-voyants pour naviguer sur le Web. Ils procèdent comme les voyants qui scannent la page avec les yeux, le regard s'arrêtant sur tout ce qui est mis en valeur visuellement (les images, ce qui bouge, ce qui est écrit en gros...) et tout ce qui est cliquable.

Or, dans cette fenêtre, seuls les intitulés de liens sont disponibles, pris hors contexte, donc. Impossible de savoir à quoi renvoie la longue liste des liens « cliquez ici » si rien ne les explicite. Bref, un lien non explicite hors contexte reste peu accessible, finalement...

Ajoutons à cela qu'il devient impossible de donner des consignes simples et claires aux personnes non techniques pour créer du contenu accessible. La définition de contexte étant entendue dans un sens programmatique, on ne peut l'évaluer si on ne lit pas le HTML.

Vous aurez compris notre scepticisme quant à l'accessibilité réelle des liens non explicites hors contexte. Notre conseil sera donc de viser l'Or pour ce cas précis et de continuer à préconiser à vos contributeurs de créer des liens explicites hors contexte, dans le sens où l'on peut comprendre leur fonction et leur destination si on les lit seuls.

L'argument d'un meilleur référencement pour des liens explicites hors contexte et de l'inutilité totale d'un lien comme « cliquez ici » devrait finir de vous convaincre.

ASTUCE Le cas des liens « lire la suite »

Certains auront peut-être bondi en lisant nos conseils qui précèdent en faveur des liens explicites hors contexte.

« Je croyais que l'accessibilité ne devait pas être un frein mais une opportunité ? » nous direz-vous... Il est vrai qu'un lien à rallonge reprenant le titre de l'article est indigeste, visuellement impossible à rendre esthétique et très lourd pour les lecteurs qui voient.

En effet, un lien comme « lire la suite » est très utile pour tronquer un texte long et tout à fait explicite pour les personnes n'ayant pas de handicap visuel. Difficile de caser un « Lire la suite de l'article [titre de l'article] » sur toute la page à chaque article tronqué.

C'est là qu'interviennent nos chers développeurs ! Car s'il est difficile pour un contributeur de rendre un lien accessible hors contexte tout le temps, la plupart des articles tronqués par un « lire la suite » ou un « en savoir plus » sont créés par des logiciels de gestion de contenus qui permettent aux contributeurs de faire cette coupure via une fonction du logiciel. Le lien est donc inséré automatiquement.

Dans la mesure où il est créé automatiquement, rien ne vous empêche alors de prévoir un lien complet, explicite hors contexte, reprenant le titre de l'article tronqué, d'ajouter une balise `<span>` pour encadrer le morceau de texte trop long et de le cacher en CSS (avec un `display:block` et un envoi loin dans la marge, à `-10000px` par exemple). Ainsi, le lien reste explicite hors contexte pour ceux qui utilisent un lecteur d'écran sans perturber l'affichage à l'écran ni encombrer vos contributeurs de règles qu'ils ne seront pas capables de suivre.

Vous pouvez, en revanche, leur apprendre à bannir les « cliquez ici ».

Le cas des images liens

L'intitulé d'un lien fait sur une image sera fourni par l'alternative de cette image. Ainsi, pour une balise `<img />` entourée de balises `<a>`, l'intitulé du lien réside dans l'attribut `alt` de la balise `<img />`.

Exemple : `<a href="..."></a>`

Il s'agit donc de rendre explicite cet attribut `alt`, en indiquant la fonction et la destination du lien. Et si l'image est en plus porteuse d'information, sachez que c'est la nature de l'image (un lien) qui est prioritaire.

Notez qu'une image-lien pour laquelle l'attribut `alt` est absent rend le critère 6.3 non applicable, cette image étant déjà invalidée dans la thématique « Images » (critère 1.1).

Les liens composites

Un lien composite est un lien qui comporte à la fois du texte et une image entre les balises `<a>` et `</a>`.

Figure 9-1

Liste de liens composites
sur le site de la ville
d'Angoulême
(<http://www.angouleme.fr>)



On voit sur la capture d'écran (figure 9-1) que le premier lien de la liste, intitulé « Festival international de la bande dessinée », contient une image. Or cette image est de la pure décoration. Elle nécessite un attribut `alt` vide. Et c'est exactement ce qui a été fait sur le site, comme on peut le voir à travers le code source du lien :

```
<a href="spip.php?mot39">

Le Festival international de la bande dessinée
</a>
```

Quand le texte est suffisamment explicite, l'attribut `alt` de l'image doit donc être vide.

En revanche, si l'image apporte une information utilisée pour expliciter le texte du lien, il faut renseigner de façon appropriée le `alt` pour que le lien soit explicite. C'est ce que nous pouvons voir sur le site de la mairie de Ruoms (figure 9-2, page suivante).

On voit sur la capture d'écran une liste de liens composés d'une image représentant le logo PDF et d'un texte avec le titre du document en téléchargement.

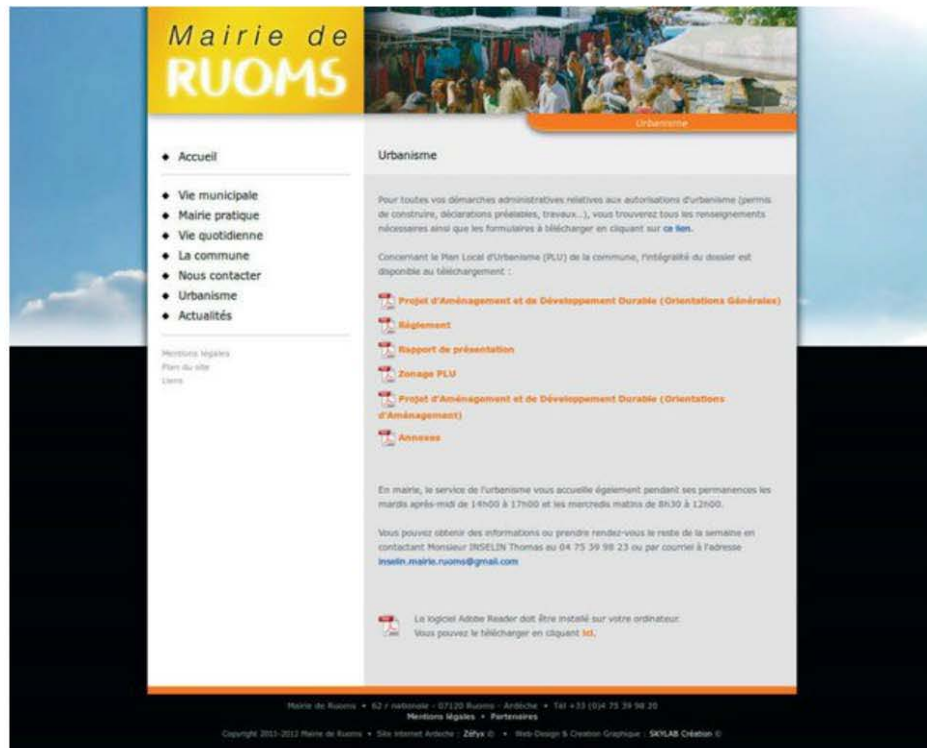


Figure 9-2 Liste de documents PDF à télécharger sur le site de la mairie de Ruoms (<http://www.ruoms.fr/urbanisme.php>) : le lien est composé d'un logo PDF et d'un texte reprenant le titre du document.

Voyons le code source du premier lien :

```
<a target="_blank" href="pdf/projet-d-amenagement-et-de-developpement-
durable-orientations-generales-FR-3.pdf">

<span class="couleur_orange">Projet d'Aménagement et de Développement
Durable (Orientations Générales)</span>
</a>
```

L'image ici n'est pas purement décorative, elle indique le format du document en téléchargement. L'attribut `alt` est pourtant vide, ce qui est une erreur. Il aurait fallut indiquer une alternative à l'image de cette façon :

```
<a target="_blank" href="pdf/projet-d-amenagement-et-de-developpement-
durable-orientations-generales-FR-3.pdf">

<span class="couleur_orange">Projet d'Aménagement et de Développement
Durable (Orientations Générales)</span>
</a>
```

Ainsi, les personnes utilisant un lecteur d'écran sauront que le lien permet le téléchargement d'un document PDF.

Halte aux attributs title abusifs !

Nous avons vu qu'un lien pouvait être explicité par son contexte. De plus, la définition du glossaire AccessiWeb précise bien qu'un attribut `title` est à considérer comme un élément de contexte. Ainsi, un lien « télécharger » sur lequel il y aurait un attribut `title` précisant « télécharger nos conditions générales de vente » rendrait le lien explicite.

Cependant, si le critère 6.1 autorise l'utilisation des attributs `title` pour apporter une information de contexte, le critère 6.2 demande à ce que chaque attribut `title` soit pertinent. Ainsi, si l'attribut n'apporte rien de plus que l'intitulé du lien, il ne doit pas être utilisé.

La bonne pratique, en l'occurrence, est d'utiliser le moins possible les attributs `title` sur les liens. En effet, certains utilisateurs de lecteurs d'écran paramètrent leur logiciel pour qu'à la fois l'intitulé et le titre du lien soient restitués. Dans notre exemple précédent cela donnerait : « [lien] Téléchargez nos conditions générales de vente ».

Alourdir la lecture pour ces utilisateurs avec des `title` inutiles rend donc l'accès à l'information plus laborieux. Assurez-vous de ne les utiliser que si c'est vraiment nécessaire.

Le cas des liens identiques

Il peut arriver qu'on soit obligé de faire deux liens avec le même intitulé. C'est le cas des très nombreux « Lire la suite » qu'on retrouve parfois, ou d'autres liens comme « Téléchargez » qu'on retrouve sur la capture de la figure 9-3.

Figure 9-3

Capture d'écran de la ville de Commentry sur laquelle on peut télécharger des documents
(<http://ur1.ca/9zrd1a>)



a. <http://www.ville-commentry.fr/decouvrir.asp?ldPage=3953>

Pour que ces liens soient considérés comme identiques, il faut les évaluer dans leur contexte. Or, les liens [Téléchargez](#) qu'on voit sur la capture font suite au titre de l'article dont il dépendent. Ils sont donc bien cadrés par un contexte qui permet de les rendre uniques. Ce ne sont pas des liens identiques.

Par contre, si cette même page n'utilisait pas correctement les balises `<h>` pour structurer la page avec des titres, les liens [Téléchargez](#) seraient devenus identiques. Dans ce cas, attention ! Les liens identiques doivent avoir la même fonction et la même destination. Si ce n'est pas le cas, le critère 6.4 est invalidé.

Notons en revanche que rien n'interdit à deux liens ayant la même fonction et la même destination d'avoir des intitulés différents.

Scripts

« Donner si nécessaire à chaque script une alternative pertinente. Rendre possible le contrôle de chaque code script au moins par le clavier et la souris et s'assurer de leur compatibilité avec les technologies d'assistance. »

<http://ur1.ca/a0cej>³

Décrié à l'origine par certains comme étant un sous-langage de programmation, impossible aujourd'hui d'imaginer un monde web sans JavaScript ! Utilisé au départ pour dynamiser les pages, son utilisation va bien au-delà désormais, concourant à créer de véritables applications. On peut même d'ailleurs dire qu'une application web est de plus en plus une application JavaScript utilisant HTML plutôt que l'inverse.

À la base des scripts, un standard : l'ECMA-262 qui a donné naissance à l'ECMAScript. Il s'agit d'un langage de programmation orienté objet. Les spécifications de l'ECMA-262 sont mises en œuvre dans différents langages de script, dont JavaScript ou ActionScript. JavaScript est un sur-ensemble d'ECMAScript développé par la fondation Mozilla et par Adobe, lui ajoutant des fonctionnalités au fil des versions.

Notons que la dernière version de HTML5 consacre JavaScript puisqu'elle intègre des modules faits pour fonctionner avec ce langage.

Plus de dynamisme, c'est bien, à condition que cela prenne en compte l'accessibilité, bien sûr. C'est bien là le défi, défi auquel des solutions comme WAI-ARIA tentent d'apporter des réponses (voir le chapitre 12, « HTML5, ARIA et accessibilité »).

JavaScript n'a jamais vraiment posé de problème d'accessibilité, les technologies d'assistance étant capables d'interpréter ce langage. Le problème tient à deux choses :

3. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#script

- Ajax : lorsqu'il y a une mise à jour dans la page sans qu'elle ne soit rechargée, les technologies d'assistance ne peuvent pas la détecter, à moins que le curseur ne soit positionné sur la zone ayant été modifiée à ce moment précis ;
- une petite proportion d'utilisateurs est contrainte de naviguer avec JavaScript désactivé. C'est particulièrement vrai dans certains réseaux d'entreprises, au sein desquels la politique de sécurité implique une désactivation de JavaScript.

Pour ces raisons, la position d'AccessiWeb 2.2 est de toujours demander une alternative à JavaScript. Cependant, avec la généralisation de HTML5 et des applications web, cette position n'est plus tenable, notamment quand il s'agit d'évaluer des processus réalisés en JavaScript. Une alternative est parfois impossible à mettre en place. Nous verrons dans le chapitre 12 ce qu'il est alors possible de faire pour contourner ce problème, grâce à WAI-ARIA.

Pour l'heure, intéressons-nous au traitement de cette thématique « Scripts » par AccessiWeb 2.2.

INFO Les critères de la thématique « Scripts »

La thématique « Scripts » contient six critères, dont cinq de niveau Bronze et un de niveau Or. Nous traiterons de tous les critères présents dans la thématique.

Alternative à JavaScript

Nous venons de dire qu'à terme, AccessiWeb comme le RGAA devront revoir leur position sur JavaScript, car demander absolument une alternative est de moins en moins pertinent aujourd'hui, à condition que le code JavaScript soit accessible.

ASTUCE L'accessibilité, c'est d'abord un choix

Beaucoup utilisent aujourd'hui des bibliothèques de composants plutôt que de tout coder eux-mêmes. Or, les bibliothèques JavaScript les plus connues proposent toutes des scripts plus ou moins accessibles, avec des modules offrant la possibilité de restituer le contenu même sans JavaScript ou intégrant de l'ARIA selon les cas. En fonction de vos besoins, pensez donc à vérifier l'accessibilité des scripts proposés. Cela vous fera gagner du temps par la suite pour un rendu plus accessible.

Cependant, alors que certains ne font plus rien sans JavaScript, sans se soucier de l'accessibilité, d'autres font des efforts pour apporter une alternative, ce qui est bénéfique pour tous ceux qui n'ont pas cette technologie activée. Ce qui prouve donc que c'est possible.

OUTILS Désactiver JavaScript

Pour désactiver JavaScript, vous pouvez utiliser la barre d'outils Web Developer Toolbar. Dans l'onglet *Désactiver*, cliquez sur *Désactiver JavaScript*. Et le tour est joué !

Autre extension Firefox pratique pour désactiver différentes technologies en un clic : QuickJava.

► <http://ur1.ca/a7iq2^a>

N'oubliez pas de réactiver JavaScript quand vous aurez terminé votre test, sinon vous risquez de vous demander pourquoi certains de vos sites favoris ne fonctionnent pas !

a. <https://addons.mozilla.org/fr/firefox/addon/quickjava/>

Prenons l'exemple du site du Conseil général du Val d'Oise. Son menu déroulant est en JavaScript (figure 9-4).



Figure 9-4 Page d'accueil du Conseil général du Val d'Oise : JavaScript est activé, le menu déroulant fonctionne. (<http://valdoise.fr>)

Quand on désactive JavaScript, le menu déroulant se transforme pour apparaître déroulé (figure 9-5). On peut ainsi accéder à tous les contenus, même quand JavaScript est désactivé. La mise en page diffère, mais tous les contenus sont accessibles, et c'est bien là l'essentiel.

Prenons un autre exemple, celui du site de l'Onisep (figure 9-6). Un large menu déroulant est lui aussi présent grâce à JavaScript.

Or quand on désactive JavaScript (figure 9-7, page 170), une grande partie des contenus devient invisible, car le menu déroulant est certes déroulé, mais de telle sorte qu'il est impossible de comprendre quelle sous-rubrique dépend de quelle rubrique.

Contrairement au site du Conseil général du Val d'Oise, l'Onisep n'a visiblement pas pensé son site pour qu'il fonctionne sans JavaScript. Bref, un site qui invaliderait à coup sûr le critère 7.2, l'alternative au script du menu déroulant n'étant pas pertinente.

Figure 9–5

Quand JavaScript est désactivé, le menu du Conseil général du Val d'Oise apparaît déroulé.

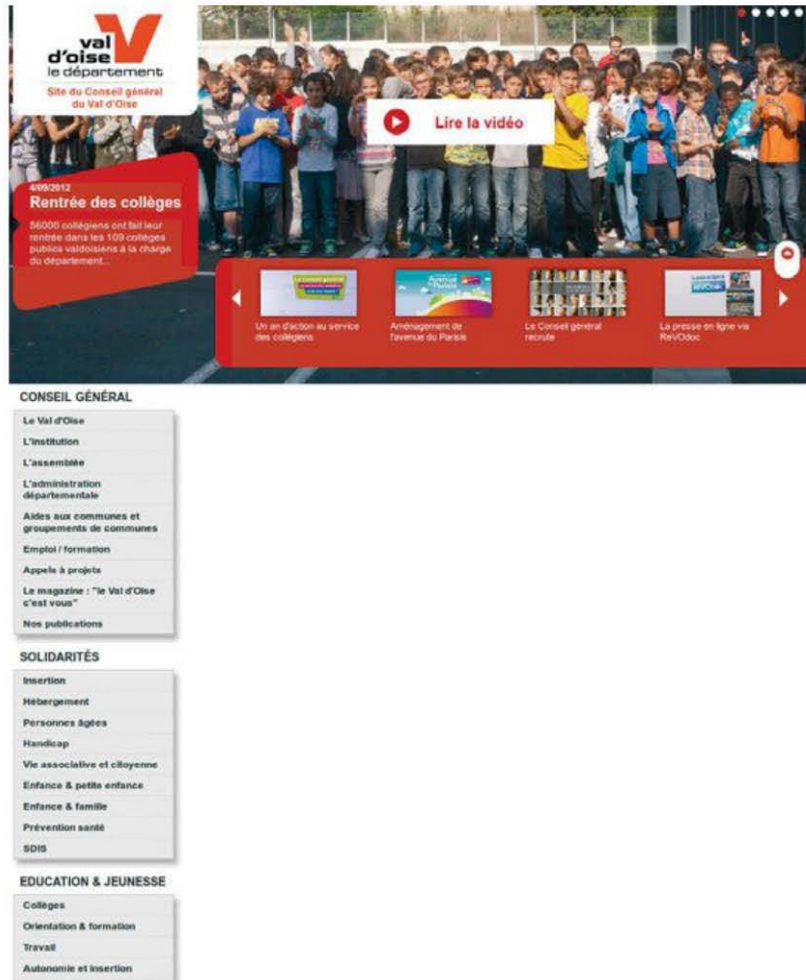


Figure 9–6

Site de l'Onisep quand JavaScript est actif : on voit un menu déroulant. (<http://onisep.fr>)



The screenshot shows the Onisep.fr website with JavaScript disabled. The page is mostly blank with a blue header and footer. The main content area is a grid of links, but most of the text is missing, leaving only the link structure visible. The header includes the Onisep logo and navigation links. The footer contains contact information and a search bar.

Figure 9-7 Le site de l'Onisep, quand JavaScript est désactivé : la majeure partie des contenus disparaît et on est incapable de se repérer dans ce qu'il reste de menu.

Pour faire une alternative, il y a plusieurs possibilités. Vous pouvez utiliser les balises `<noscript>` et `</noscript>`. Elles permettent de charger le contenu en cas de désactivation de JavaScript, tout en le laissant invisible quand JavaScript est actif. Ceci dit, cet usage est limité puisque vous ne pouvez utiliser ces balises qu'une seule fois, alors que le code JavaScript est souvent disséminé dans toute la page. Par ailleurs, `<noscript>` est proscrite en XHTML. Enfin, il arrive que les scripts soient actifs, mais que pour une raison inconnue, leur exécution échoue. Dans ce cas, les scripts étant actifs, le contenu de la balise `<noscript>` ne sera pas rendu. C'est pourquoi il vaut mieux éviter de recourir à cette balise et concevoir plutôt ses scripts pour qu'ils modifient la page d'un état acceptant les scripts à un état sans script au moment de l'exécution.

Vous pouvez ainsi prévoir une alternative directement dans la page lorsque JavaScript est désactivé. C'est ce que nous avons pu observer sur le site du Conseil général du Val d'Oise.

L'alternative peut également être donnée via un lien permettant d'accéder à une version alternative de la page. Bien que ce ne soit pas idéal dans une optique de conception universelle, dans certains cas particulièrement complexes, vous n'aurez sans doute pas d'autre solution que de fournir une page alternative fonctionnant sans JavaScript.

Enfin, il est possible de prévoir une alternative via un langage de script côté serveur.

Pensez également à fournir une alternative pour les scripts déclenchant l'ouverture d'une nouvelle fenêtre, et à faire en sorte que l'alternative soit mise à jour de la même façon que le script, le cas échéant. Une alternative pertinente est celle qui permet d'*accéder aux fonctionnalités et aux contenus similaires* à ceux proposés dans la version par défaut en JavaScript.

MÉTHODE Tester la présence de JavaScript

Voici la méthodologie de test proposée par BrailleNet pour tester les critères relatifs à JavaScript. La première partie est commune à tous les critères.

Avec la barre d'outils Web Developer pour Firefox :

1. Dans le menu *Information*, activez l'option *Voir le JavaScript* pour vérifier l'existence d'un fichier JavaScript associé à la page.
2. Dans le menu *Voir source*, activez l'option *Voir le code source généré* et chercher (CTRL+F) « JavaScript » pour vérifier l'existence d'événements JavaScript sur des liens.
3. Cherchez les termes suivants pour vérifier l'existence d'événements JavaScript sur des éléments HTML : *onBlur*, *onChange*, *onClick*, *onDbclick*, *onDragdrop*, *onFocus*, *onKeyDown*, *onKeyPress*, *onLoad*, *onMouseover*, *onMouseout*, *onSelect*, *onSubmit*, *onUnload*, *onResize*.
4. Repérez dans la page les événements concernés.
5. Dans le menu *Désactiver*, choisissez l'option *Désactiver JavaScript* puis *Tout le JavaScript*.

Puis, selon les critères, il s'agira de vérifier s'il y a bien une alternative dans la page, via une balise `<noscript>` ou via un lien, si cette alternative est pertinente, etc.

Notons que le test 7.1.2 demande également de vérifier la présence du terme `window.open` dans la page pour vérifier l'existence d'une commande JavaScript d'ouverture de nouvelle fenêtre.

Cette méthodologie est relativement complexe à mettre en œuvre. Pour un usage courant, il n'est pas forcément nécessaire d'y recourir. Il vous suffira d'*observer la page avec JavaScript actif, puis de comparer les comportements quand JavaScript est désactivé*. Cela suffit généralement à traiter cette thématique. Mais en cas de doute, vous pouvez toujours recourir à la méthodologie de test, lourde mais exhaustive, proposée par BrailleNet.

Une alternative « si nécessaire »

Les critères relatifs à une alternative à JavaScript précisent bien que cette alternative doit être fournie « *si nécessaire* ». Dans quel cas cela n'est-il pas nécessaire ?

Eh bien, lorsque JavaScript n'est utilisé que pour améliorer l'expérience utilisateur, en produisant des effets particuliers mais sans modifier les contenus. Dans ce cas, quand on le désactive, les contenus restent inchangés. C'est le cas également pour des scripts qui n'ont pas vocation à être restitués, comme les scripts utilisés pour recueillir les statistiques de fréquentation.

Une page web est composée de plusieurs couches. La couche de base qui concerne la structure et les contenus, c'est HTML. Puis vient la mise en page, pour choisir les couleurs, la taille des textes, leur positionnement... Ce sont les feuilles de styles CSS qui s'en chargent. Enfin, pour faire un effet de transition douce sur une ancre par exemple, ou apporter un mouvement au survol d'un élément, on peut utiliser JavaScript.

Penser son site en pensant aux contenus, puis à la mise en page et en dernier aux effets garantit sans doute une plus grande facilité pour rendre son site accessible, car le contenu ne dépend alors ni des CSS, ni de JavaScript. Tommy Olsson, dans un article du 7 février 2007, parlait d'*amélioration progressive*.

De même, pour parler du fait que son site fonctionne, même avec une mise en page dégradée quand CSS et JavaScript sont désactivés, Tommy Olsson évoquait une *dégradation élégante*.

RESSOURCE Dégradation élégante et amélioration progressive

Retrouvez en ligne l'article de Tommy Olsson du 6 février 2007.

Article original en anglais :

▶ <http://ur1.ca/a4irw>^a

Traduction en français sur le site de Pompage :

▶ <http://ur1.ca/a4irs>^b

a. <http://accessites.org/site/2007/02/graceful-degradation-progressive-enhancement/>

b. <http://www.pompage.net/traduction/degradation-elegante-et-amelioration-progressive>

Amélioration progressive et dégradation élégante. Ce principe simple, mis en avant il y a déjà cinq ans, préconise donc de ne pas faire reposer l'organisation de sa page et la restitution de ses contenus sur une technologie à laquelle tout le monde n'a pas forcément accès.

Mais cela est-il vraiment possible dans le cas de certaines applications web complexes reposant sur JavaScript ? Et quand bien même ce serait possible, au prix de combien d'efforts et de contournements ? Bref, sachant que les technologies d'assistance interprètent JavaScript, peut-on encore considérer aujourd'hui que cette technologie nécessite forcément une alternative ? Le débat est lancé...

À NOTER Les WCAG autorisent JavaScript, s'il est accessible

Notons que les WCAG 2 n'interdisent pas JavaScript, mais demandent seulement que les scripts soient « compatibles avec l'accessibilité » et qu'une alternative soit fournie si ce n'est pas le cas. Or, les référentiels AccessiWeb comme RGAA ont la double exigence d'une alternative et d'une compatibilité avec les technologies d'assistance.

Cette position est à la fois plus exigeante que celle des WCAG 2 et plus limitative, car elle restreint de fait l'utilisation de WAI-ARIA à quelques applications dont la compatibilité est vérifiée, comme l'attribut `tabindex` pour étendre la prise en charge du focus à un élément quelconque. Cette position devrait évoluer assez rapidement. On peut penser qu'une évolution du référentiel AccessiWeb allant dans le sens des WCAG 2 voie le jour courant 2013, dès lors que la prise en charge de WAI-ARIA sera généralisée. Pour ce qui est du RGAA, par contre, tant qu'aucune volonté politique ne permettra un suivi et un accompagnement du référentiel, il restera sur le bord du chemin des évolutions web...

Le cas particulier de l'environnement maîtrisé

Nous abordons pour la première fois ce concept d'*environnement maîtrisé*, mais c'est un des apports d'AccessiWeb et du RGAA pour contextualiser davantage l'application des règles. Il s'agit de faire du cas par cas et de prendre en compte les contextes dans lesquels on connaît précisément le cadre technique de l'utilisateur.

Ainsi, dans le cadre d'une entreprise, par exemple, au sein de laquelle vous connaissez le système d'exploitation, la version du navigateur installée et les technologies actives de tout le personnel, on parle d'environnement maîtrisé. Vous maîtrisez en effet avec certitude le cadre technologique qui va permettre aux personnes de l'entreprise de naviguer sur le Web. Vous pouvez donc vous assurer que tout le monde a JavaScript actif sur son poste et dispose, le cas échéant, d'une technologie d'assistance capable de restituer le code JavaScript. Vous n'êtes alors pas tenus de fournir une alternative à JavaScript. Il s'agit d'un cas particulier rendant les critères 7.1 et 7.2 non applicables.

Un script accessible

Autant nous discutons de la pertinence de demander systématiquement une alternative à JavaScript, autant les critères 7.3 et 7.4 sont indiscutablement très importants, puisqu'ils demandent respectivement à ce que le script soit accessible au clavier et à la souris et « compatible avec les technologies d'assistance ». En bref, *il faut que le script soit accessible*.

Le glossaire AccessiWeb précise que « le contrôle d'une fonctionnalité par le clavier signifie que celle-ci est accessible via la touche *Tabulation* et activable par la touche *Entrée* à l'exception des contrôles complexes (slider, bouton rotatif...) qui peuvent utiliser des touches dédiées comme les touches de direction. » (<http://ur1.ca/a7lv>⁴)

Notons que cette obligation d'utiliser uniquement les touches *Tabulation* et *Entrée* va sans doute évoluer, quand il sera possible d'appliquer les recommandations de WAI-ARIA en termes de conception (les *design patterns*, voir <http://ur1.ca/a8exy>⁵). Ces recommandations préconisent l'utilisation des touches du contrôle au clavier issues des modèles logiciels : *Esc*, *Tab*, *Space* et les touches de direction. Ce jeu de touches est déjà utilisé sur certains composants d'interfaces natifs comme les listes déroulantes, via la balise *select* par exemple. Il est probable que la définition du glossaire AccessiWeb change pour intégrer ce jeu de touches en 2013, quand AccessiWeb jugera l'implémentation d'ARIA suffisante.

RGAA Raccourcis clavier autorisés

Notons que le RGAA autorise déjà l'utilisation de raccourcis clavier comme alternative à l'utilisation de *Tabulation+Entrée*. Il se différencie sur ce point d'AccessiWeb.

Mais il faut également que le rôle, l'état, les paramétrages et les changements soient restitués (ce qui nécessitera souvent l'utilisation de WAI-ARIA – voir le chapitre 12), ou bien qu'une alternative accessible avec des fonctionnalités équivalentes soit présente.

Navigation au clavier et à la souris

Pour tester la navigation au clavier et à la souris, il vous faudra vérifier que les éléments disposant d'un gestionnaire d'événements soient accessibles au clavier et à la souris ou qu'une action similaire est disponible et accessible en alternative. L'activation doit pouvoir se faire via la touche *Entrée*.

4. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#mControleClavSour

5. http://www.w3.org/TR/wai-aria-practices/#aria_ex

Pour les contrôles complexes tels que des diaporamas ou des jauges, il vous faudra vérifier qu'il existe des touches du clavier permettant de les activer. Enfin, non seulement le JavaScript doit être accessible ou fournir une alternative accessible, mais il faut également s'assurer que les éléments HTML qui reçoivent nativement le focus (les liens et les contrôles de formulaires) peuvent toujours recevoir le focus quand JavaScript est activé.

Des scripts compatibles avec les technologies d'assistance

Concernant la compatibilité avec les technologies d'assistance, rien de tel qu'un test de la page avec un lecteur d'écran, comme NVDA ou Orca. L'objectif est de s'assurer que l'utilisateur pourra interagir avec les composants d'interfaces générés ou contrôlés par JavaScript.

Seuls les liens (balise `<a>`) et les contrôles de formulaires (balises `<input>`, `<select>`, `<button>`, `<textarea>`) sont prévus pour recevoir le focus en HTML.

CONSEIL Privilégier la balise `<button>` à la balise `<a>`

Notons qu'il vaut mieux utiliser la balise `<button>` à la balise `<a>` pour implémenter des fonctionnalités en JavaScript, et ce pour plusieurs raisons :

- c'est l'élément dédié par HTML pour créer des boutons d'action dans le code, par exemple pour les boutons *Suivant/Précédent* d'un diaporama ;
- cela évite d'avoir des balises `<a>` détournées de leur usage avec un attribut `href` inutile ;
- cela permet d'utiliser le type `button` et donc de faire en sorte que les lecteurs d'écran annoncent « bouton » au lieu de « lien » ;
- cela permet l'emploi de l'attribut `disabled` pour signaler que le bouton est inactif, information qui sera vocalisée par les lecteurs d'écran ;
- contrairement aux liens, on peut mettre dans l'élément `button` tous les types de contenus, image ou texte.

Pour en savoir plus sur cette balise `<button>`, voir le site du W3C :

► <http://ur1.ca/a8ffe>^a

a. <http://dev.w3.org/html5/markup/button.html>

S'il est possible avec WAI-ARIA de générer ou de créer des composants d'interfaces avec d'autres éléments, AccessiWeb 2.2 précise :

« Les composants d'interfaces générés et créés par JavaScript en utilisant des éléments HTML, autres que des liens ou des contrôles de formulaires, nécessitent l'utilisation d'ARIA pour être rendus accessibles. L'implémentation d'ARIA n'étant pas définitive, ces cas de figures sont jugés non-conformes, actuellement. »

Après avoir vérifié que les composants d'interfaces sont bien des liens ou des contrôles de formulaires, vous devrez vous assurer que, le cas échéant, le nom de chaque composant (intitulé, balise `<label>` ou attribut `title`) est bien mis à jour en fonction de l'action réalisée. Si ce n'est pas le cas, une version alternative accessible devra être fournie, sous peine d'invalider le critère 7.4.

Les changements de contexte

Attention, les changements de contexte sont un sujet complexe !

Une définition complexe

Imaginez... Vous naviguez tranquillement sur une page qui vous permet d'acheter différentes variétés de thé. Vous vous apprêtez à acheter votre thé, mais avant, vous cliquez sur un lien qui vous intrigue : « Thé à gogo ». Et là, vous vous retrouvez sur un jeu concours, et impossible de retrouver votre panier, dans lequel vous aviez mis le thé que vous souhaitiez acheter. Vous cherchez sur la page, mais elle n'a plus rien à voir avec celle sur laquelle vous naviguiez, il y a deux secondes à peine.

Que s'est-il passé ? Vous avez changé de page en cliquant malencontreusement sur une publicité (cachée dans la page et activée via JavaScript), mais comme vous n'avez pas été averti et comme vous ne voyez pas la page dans son ensemble, vous êtes perdu !

Ce problème vous concerne également si vous êtes aveugle ou atteint d'un trouble de la vue vous empêchant de voir la page dans son entier : si vous ajoutez un produit dans un panier mais que rien ne vous indique que le panier a été mis à jour, vous risquez d'ajouter plusieurs fois le même article sans vous en rendre compte !

Ainsi, les changements de contexte sont caractérisés par des changements majeurs dans le contenu de la page qui, s'il sont faits sans que l'utilisateur n'en soit conscient, peuvent le désorienter.

Un changement de contexte, dans l'absolu, est assez facile à imaginer. C'est un concept générique qui peut s'appliquer à un grand nombre de critères des WCAG, comme le téléchargement de fichiers ou de médias, les liens explicites, la pertinence des intitulés des boutons de soumission, l'ouverture d'une nouvelle fenêtre, l'indication du focus, les changements d'état des intitulés, ou encore les textes cachés en CSS.

Or, la notion introduite par AccessiWeb est assez limitative, puisqu'elle a été pensée pour traiter les cas *ne pouvant pas déjà être traités par d'autres thématiques*, telles la thématique 6 sur les liens ou la thématique 12 sur la navigation. AccessiWeb a d'abord cherché à traiter les changements de contexte de façon transparente, au travers des autres thématiques. Mais comme JavaScript permet de simuler de nombreux comportements, les rédacteurs n'ont pas réussi à supprimer cette notion du référentiel.

La définition du changement de contexte donnée dans le glossaire AccessiWeb précise (<http://ur1.ca/a7uki>⁶) :

- « Les changements de contexte comprennent les changements :
 - d'agent utilisateur
 - d'espace de restitution
 - de focus
 - de contenu qui modifie la signification de la page web »

Et de prendre les exemples suivants comme caractéristiques d'un changement de contexte : « l'ouverture d'une nouvelle fenêtre, le déplacement du focus sur un composant différent, le déplacement vers une nouvelle page (y compris tout ce qui, pour l'utilisateur, aurait l'air d'un déplacement vers une autre page) ou la réorganisation significative du contenu d'une page ».

Notons qu'un changement de contenu n'est pas forcément un changement de contexte. Le déploiement d'une arborescence, un menu dynamique ou un déplacement de tabulation ne changent pas nécessairement le contexte, à moins qu'il ne change aussi l'un des éléments énumérés dans le glossaire (le focus, par exemple).

La définition d'un changement de contexte dans le cadre d'AccessiWeb est donc très difficile, puisque cette notion est là pour prendre en charge des situations « inconnues » par définition.

Cela se double d'une seconde difficulté, puisque les réparations du critère 12.13 sur l'ordre de tabulation sont susceptibles de créer une situation de changement de contexte.

Pour faire simple, retenons que *seuls les changements de contexte introduits par des scripts sont applicables*. Plus concrètement, nous verrons que cela concerne essentiellement deux types de comportement :

- les mises à jour dynamique de contenus et/ou de présentation ;
- les manipulations de composants.

Rendre un changement de contexte accessible

Pour qu'un changement de contexte soit accessible, il y a deux possibilités, au choix :

- l'utilisateur doit être averti de ce changement avant qu'il ne se produise, par un texte explicite ;
- ou le choix doit être laissé à l'utilisateur d'activer ou non ce changement, en cliquant sur un bouton de formulaire, par exemple, ou un lien explicite.

Prenons quelques exemples pour mieux comprendre de quoi il s'agit.

6. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#mChangContexte

La carte sur la page d'accueil du site de l'Institut national des langues et civilisations orientales (Inalco) est en JavaScript (figure 9-8). On peut sélectionner sur la carte la zone géographique correspondant à la langue qui nous intéresse. (Notre exemple reste un peu théorique dans la mesure où la carte n'est pas accessible au clavier. Mais cela n'a pas d'importance pour nous aider à définir un changement de contexte.)

Figure 9-8

Page d'accueil de l'Inalco :
quand on survole la carte, on
peut sélectionner une zone.
(<http://www.inalco.fr>)



Une fois qu'on clique sur la zone choisie, la page se met à jour sans être rechargée, en proposant de nouveaux liens correspondant à la sélection (figure 9-9). Il s'agit bien d'un changement de contexte. Et l'utilisateur risque de ne pas savoir qu'un nouveau contenu est apparu suite à son action.

Une solution possible serait de transporter le focus sur la zone mise à jour, en modifiant éventuellement l'intitulé de la zone cliquable de la carte pour que l'utilisateur comprenne bien ce qui se passe. Notez qu'il est inutile de signaler qu'il s'agit d'un lien interne, les lecteurs d'écran restituent cette information.

En sortie de tabulation, n'oubliez pas de redonner le focus sur la carte, à l'endroit du dernier lien activé, pour éviter d'invalider le critère 12.14 concernant la cohérence de l'ordre de tabulation.

Prenons un autre exemple de changement de contexte : la mise à jour dynamique de valeurs de champs de formulaire, ainsi qu'un comportement proche, à savoir l'insertion de champs de formulaire conditionnés à une option préalable.

Figure 9-9

Quand on clique sur la région du Moyen-Orient, une liste de liens apparaît avec les langues enseignées pour cette zone.



Ainsi, imaginons que le choix de la zone sur la carte prise dans l'exemple ci-dessus ait été proposé via une liste déroulante. Quand on a choisi la zone, la liste des langues se met à jour. Encore une fois, l'utilisateur peut être désorienté par l'activation d'un nouveau champ (si la liste des langues était cachée jusque là) ou par le changement de la valeur des éléments de la liste, si la mise à jour se fait dans la même liste déroulante.

Une solution possible consisterait à entourer les champs d'une balise `<fieldset>` avec une légende pertinente (par exemple : `<legend>Choisir la zone géographique puis la langue</legend>`) ou quelque chose d'équivalent. L'utilisateur est prévenu, le critère 7.5 est validé.

Autre solution possible : conditionner la mise à jour de la liste des langues à une action de l'utilisateur. On peut par exemple demander à l'utilisateur de choisir la zone, puis de cliquer sur un bouton pour définir la langue. L'utilisateur peut contrôler le changement de contexte, le critère 7.5 serait, là encore, validé.

Prenons un dernier exemple avec le site de courses en ligne Auchandirect.fr (figure 9-10, page suivante).



Figure 9-10 Sélection d'une tablette de chocolat dans le panier sur le site Auchandirect.fr : une zone en bas à droite de l'écran est mise à jour.

Nous voyons sur la capture précédente que nous avons sélectionné un article, du chocolat. La zone en bas à droite de l'écran se met à jour automatiquement, via JavaScript, mais le focus n'est pas déplacé sur cette zone. Cela donne l'impression que rien ne s'est passé. L'utilisateur aveugle ou ne voyant pas toute la page peut se demander si sa sélection a fonctionné et être tenté de sélectionner à nouveau l'article, ce qui va augmenter le nombre d'articles dans son panier à son insu.

ATTENTION Complexité de la définition des changements de contexte

Nous ne vous avons pas caché que définir un changement de contexte prêtait souvent à débat, tant il est parfois difficile de dire si le contenu a suffisamment été modifié pour changer la nature de la page... ou pas. Lors du séminaire AccessiWeb où la notion a été discutée, parmi les exemples présentés, les avis étaient très tranchés : environ la moitié de la salle, pourtant constituée d'experts formés, qualifiait l'exemple de changement de contexte, alors que l'autre moitié s'opposait à cette qualification.

Dans l'exemple que nous avons pris concernant la mise à jour en temps réel de la quantité de produits achetés sur le site Auchandirect.fr, nous pourrions aussi nous interroger. S'agit-il d'un problème de changement de contexte, ou plutôt d'un problème de placement du focus ? Le changement de focus est traité par le critère 12.13 concernant l'ordre de tabulation. Globalement, chaque fois qu'on manipule le focus ou qu'on met à jour des éléments de la page (formulaire ou zones), on est fortement susceptible de créer une problématique de changement de contexte.

En attendant, au-delà des querelles d'experts, retenez qu'il faut permettre à l'utilisateur de savoir ce qui se passe dans la page, quel que soit son mode de restitution ou de navigation.

Pour corriger ce problème, une possibilité (parmi d'autres) aurait été de modifier le focus en JavaScript après la saisie de l'article, en jouant sur la méthode JavaScript `focus` et les attributs `tabindex` pour aller directement dans la zone mise à jour. Puis on aurait refait pointer le focus sur l'article suivant juste après consultation (pour en savoir plus sur les attributs `tabindex`, voir l'encart page 301 dans le chapitre 12).

Autre possibilité : diriger l'internaute vers une page lui indiquant que son article a bien été ajouté en lui proposant de valider son panier ou de poursuivre ses achats.

Contrôler les alertes non sollicitées

Nous passerons rapidement sur ce critère 7.6, de niveau Or. Il s'agit ici de laisser le contrôle à l'utilisateur des alertes non sollicitées. Vous savez, ces alertes concernant la dernière promo du moment qui prend toute la page et qu'on a parfois du mal à fermer sans cliquer sur la publicité associée... Oui, ce genre d'alertes. Pas très accessible, car très perturbateur.

L'utilisateur devrait toujours pouvoir choisir d'activer ou non ce genre d'alertes via un bouton de formulaire ou un lien. Mais l'alerte, pour le coup, perd alors son caractère « non sollicité ». Ce que personne ne regrettera...

Notons que dans le cas d'une alerte exigeant une action immédiate de l'utilisateur dans une situation d'urgence, afin de préserver sa santé, sa sécurité ou sa propriété, le critère est bien entendu non applicable.

Éléments obligatoires, structuration de l'information et présentation de l'information (thématiques Accessiweb 8 à 10) 10

Les thématiques AccessiWeb numéro 8 à 10 concernent les éléments HTML obligatoires pour que le site soit accessible (8), la structuration de l'information dans la page HTML (9) et la présentation de l'information, grâce notamment aux CSS (10).

Éléments obligatoires

« Vérifier que chaque page web a un code source valide selon le type de document, un titre pertinent et une indication de langue par défaut. Vérifier que les balises ne sont pas utilisées uniquement pour la présentation, que les changements de langues et de direction de sens de lecture sont indiqués. »

<http://ur1.ca/a0cer>¹

Nous avons vu que pour qu'une page web fonctionne correctement, il faut que tout le monde joue le jeu : les développeurs de navigateurs doivent intégrer la spécification HTML ; les développeurs de technologies d'assistance doivent prendre en compte les éléments et attributs pour les restituer ; et enfin, les auteurs (concepteurs, développeurs et contributeurs web) doivent concevoir des contenus web accessibles.

Bref, tout repose sur une spécification commune, le HTML. Il suffit qu'un des maillons ne joue pas le jeu pour que l'accessibilité soit compromise.

Il est donc important de connaître et de respecter certains éléments obligatoires. Il s'agit d'éléments techniques, ceux qui constituent la grammaire du Web. Car comme tout langage, le HTML a une grammaire propre dont certains éléments sont expres-

1. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#elements

sément prévus pour plus d'accessibilité – que ce soit au niveau de la structure de la page ou à l'intérieur du document. Voyons cela en détail.

INFO Les critères de la thématique « Éléments obligatoires »

La thématique « Éléments obligatoires » contient dix critères, dont huit de niveau Bronze et deux de niveau Argent. Nous traiterons de tous les critères présents dans la thématique.

Sécuriser l'accès des technologies d'assistance à la page web

Il y a deux grands types de règles pour s'assurer que les technologies d'assistance peuvent accéder à la page et en identifier le contenu sans erreur : les règles qui permettent de savoir comment est faite la page (version de la spécification HTML, titre, langue...) et celles qui visent à éviter une mauvaise interprétation du code en donnant de fausses informations.

Définir un cadre technique clair

Rappelons en préambule de quoi est composée une page web. Elle contient des éléments : les *balises* HTML, elles-mêmes pouvant être dotées d'*attributs*.

Le HTML est le nom générique d'un langage qui a beaucoup évolué et continue à évoluer. Cela a donné lieu à plusieurs versions de ce même langage. AccessiWeb 2.2 est basé sur HTML4, mais il existe une version de HTML encore en développement et déjà partiellement implémentée : HTML5 (voir le chapitre 12, « HTML5, ARIA et accessibilité »).

Au cours de l'évolution du Web, la spécification HTML a changé : certains éléments et attributs ont été créés pour répondre à de nouveaux besoins, d'autres ont été bannis de la spécification (on dit qu'ils sont obsolètes) ; d'autres, enfin, ont connu une phase transitoire en étant dépréciés.

Une balise ou un attribut est *déprécié* quand il est remplacé par une balise ou un attribut plus récent. Les éléments dépréciés sont susceptibles de devenir obsolètes dans les versions futures de HTML. Le W3C indique clairement que les navigateurs et les technologies d'assistance doivent continuer à prendre en charge les éléments dépréciés pour des raisons de rétrocompatibilité.

En revanche, quand un élément est *obsolète*, il disparaît de la spécification et devient interdit. Par exemple, les attributs HTML de présentation ont été dépréciés quand leurs alternatives en feuilles de styles sont apparues. Une phase de transition était donc encore permise. Mais tous les éléments dépréciés en HTML4 sont devenus obsolètes en HTML5.

Par ailleurs, il existe des balises et des attributs inconnus de la spécification HTML. On parle d'*éléments propriétaires*.

Dans les années 1990, pendant les débuts du Web, l'imagination des concepteurs web était bridée par une spécification qui ne permettait pas de faire beaucoup d'effets visuels. Pour répondre à ce besoin, les développeurs de navigateurs ont ajouté unilatéralement de nouvelles fonctionnalités à HTML en dehors des standards de l'époque. Ce qui signifiait que de nouveaux éléments HTML propriétaires fonctionnaient dans certains navigateurs et pas dans d'autres.

L'évolution du HTML et l'introduction des CSS en 1998 a ralenti cette tendance. De nouvelles constructions sémantiques ont permis de remplacer certains de ces attributs propriétaires. Ainsi, la balise `object` visait à remplacer les balises propriétaires `applet` et `embed`. Certaines balises propriétaires ont parfois même été standardisées : c'est le cas de `embed` qui a été officiellement intégrée dans HTML5.

Ce qu'il faut retenir, c'est que *chaque version de HTML a ses règles propres*, avec son lot de balises et d'attributs autorisés et interdits. Il est important de bien connaître ces règles pour écrire un code qui soit utilisable correctement à la fois par les navigateurs et les technologies d'assistance. C'est la base première pour permettre une restitution correcte de ces éléments aux utilisateurs finaux.

RESSOURCES Éléments dépréciés, obsolètes et propriétaires

Pour connaître la définition précise (en anglais) de « déprécié » et « obsolète », voir le site du W3C :

- ▶ <http://ur1.ca/a4vvo^a> (déprécié)
- ▶ <http://ur1.ca/a4vwg^b> (obsolète)

Notons que tous les éléments seulement dépréciés en HTML4 sont obsolètes en HTML5. Mieux vaut donc se reporter à la liste des éléments obsolètes en HTML5 :

- ▶ <http://ur1.ca/a4vx2^c> (éléments HTML5 obsolètes)
- ▶ <http://ur1.ca/a4vy3^d> (attributs HTML5 obsolètes)

Attention : il y a une différence en HTML5 entre les éléments obsolètes mais conformes, et les éléments obsolètes non conformes. Pour plus d'informations à ce sujet, voir :

- ▶ <http://ur1.ca/a6wf2^e>

Enfin, pour un aperçu de l'histoire des balises propriétaires, vous pouvez vous reporter à la page Wikipédia (en anglais) :

- ▶ <http://ur1.ca/a58az^f>

a. <http://www.w3.org/TR/html4/conform.html#deprecated>

b. <http://www.w3.org/TR/html4/conform.html#didx-obsolete>

c. <http://www.w3.org/TR/html5-diff/#obsolete-elements>

d. <http://www.w3.org/TR/html5-diff/#obsolete-attributes>

e. <http://dev.w3.org/html5/pf-summary/obsolete.html>

f. http://en.wikipedia.org/wiki/Tag_soup#Use_of_proprietary.2Fdiscontinued_elements

Le Doctype, grammaire du Web

Le **DOCTYPE**, appelé encore « type de document », est un élément qui introduit la déclaration de la version du langage utilisée dans la page.

Comme nous venons de le voir, il n'y a pas qu'une version de HTML. Toutes ces versions de HTML sont définies dans leur spécification W3C respective, mais elles sont aussi définies dans un langage interprétable par les machines qui spécifie la structure standard, les éléments et les attributs de la version du HTML en question.

Cette définition est appelée « définition du type de document » ou, en abrégé, DTD.

Certains outils ont besoin de connaître quelle DTD le document HTML utilise. C'est le cas des navigateurs qui, s'ils n'ont pas de type de document déclaré, utiliseront une DTD propriétaire avec un renvoi du code selon une interprétation propre. Les technologies d'assistance utilisant le code généré par les navigateurs, il y a un risque pour que le code ne soit pas interprété comme il le faudrait. La présence d'un **DOCTYPE** garantit que le navigateur utilisera la bonne spécification.

Toute page web doit donc commencer par déclarer le type de document utilisé. C'est le tout premier élément rédigé. Prenons deux exemples de DTD.

Le premier indique que la page sera conçue selon la spécification XHTML 1.0 transitionnelle. C'est une spécification que l'on retrouve assez fréquemment.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Dans notre deuxième exemple, voyons la syntaxe de la DTD relative à HTML5 :

```
<!DOCTYPE HTML>
```

Épurée, n'est-ce pas ? Cette DTD simplifiée est inhérente au principe de HTML5 qui se veut un langage en évolution.

RESSOURCE Choisir son Doctype

Vous pouvez consulter un article très complet de Laurent Denis sur le site d'Alsacréations et intitulé « Les DTD HTML4.01, XHTML1.0 et HTML5 : quel Doctype choisir ? » :

► <http://ur1.ca/9zu1j^a>

a. <http://www.alsacreations.com/article/lire/560-dtd-doctype-html-xhtml-comment-choisir.html>

« Alors, me demanderez-vous, quelle est le meilleur Doctype à utiliser pour faire de l'accessibilité ? » Eh bien, la seule bonne réponse est : « Celle qui correspond à vos besoins. »

Il n'y a pas de Doctype meilleur qu'un autre pour faire de l'accessibilité. AccessiWeb demande seulement de déclarer un type de document et de s'y conformer, en faisant un code valide selon le type de document déclaré.

OUTILS Le validateur du W3C

Pour tester la validité de votre page selon le Doctype déclaré, vous pouvez utiliser le validateur en ligne du W3C :

► <http://validator.w3.org>

Il suffit d'entrer l'adresse de la page à tester, ou de copier-coller le code si la page n'est pas encore consultable en ligne, et de cliquer sur le bouton [Check](#).

Si le validateur détecte une ou plusieurs erreurs, il vous l'indique en rouge, en mentionnant très précisément où se trouvent les erreurs, avec une explication (en anglais). Si votre page ne contient pas d'erreur, le validateur vous félicite et vous propose une série de logos pour afficher que votre page est conforme à la spécification déclarée.

Vous envisagiez de travailler en XHTML 1.0 Strict, mais vos contributeurs auront besoin d'ajouter des `<iframe>` ? Les `<iframe>` étant interdites en XHTML 1.0 Strict, optez plutôt pour un Doctype XHTML 1.0 Transitionnel.

Cette idée est valable dans l'absolu, mais en pratique, il faut aussi tenir compte des évolutions des navigateurs qui ont tendance à implémenter les dernières spécifications, tout en ne maintenant la rétrocompatibilité que dans une certaine mesure. Ainsi, en théorie, si vous ne savez pas utiliser les feuilles de styles et souhaitez utiliser des balises servant à la présentation, vous pourriez déclarer un Doctype HTML 2.0, mais celui-ci ne sera pas reconnu. Cette DTD ne fait pas partie de celles recommandées par le W3C.

RESSOURCE Les DTD recommandées par le W3C

Le W3C édite une liste des DOCTYPE recommandés, à consulter sans hésiter :

► <http://ur1.ca/3ss83>^a

a. <http://www.w3.org/QA/2002/04/valid-dtd-list.html>

En pratique, si le validateur du W3C ne rapporte aucune erreur, le critère 8.2 est validé. En revanche, s'il détecte une ou plusieurs erreurs, le critère ne sera pas forcément invalidé.

L'utilisation de balises ou d'attributs obsolètes est explicitement interdite par le W3C. Cela entraîne donc une invalidation du critère 8.2. Rien n'est dit cependant concernant les balises et attributs propriétaires. Cela provoque une invalidation par le validateur du W3C, qui ne reconnaît pas les éléments puisque ne figurant pas dans sa spécification. Mais AccessiWeb (qui applique WCAG) demande simplement à vérifier que cela ne pose pas de problème en termes d'accessibilité.

Ainsi, la balise `<embed>` ne pose pas de problème. De même pour les microformats disposant d'un schéma (à condition qu'ils ne viennent pas perturber les technologies d'assistance). Quant aux attributs applicatifs, ils sont simplement ignorés et ne sont donc pas problématiques non plus.

Enfin, une exception est accordée pour les attributs ARIA qui, s'ils sont appliqués sur un document ayant une DTD antérieure à HTML5, provoquent des erreurs avec le validateur. Mais dans la mesure où cela est fait pour améliorer l'accessibilité, AccessiWeb n'invalide pas ces attributs.

IDÉE REÇUE Ma page passe le validateur du W3C, elle est accessible !

Cette idée reçue est assez fréquemment évoquée. Bien évidemment, si vous avez déjà lu le livre jusque là, vous êtes bien conscient que passer le validateur du W3C avec succès est très loin d'être suffisant pour dire que la page est accessible. Il ne s'agit en effet que des critères 8.1 et 8.2, soit deux critères AccessiWeb sur les 133 que compte le référentiel. Un code qui respecte les standards est toutefois essentiel pour permettre sa bonne interprétation par les technologies d'assistance.

La langue du document est identifiée

Un lecteur d'écran, bien qu'étant une technologie de pointe rendant de grands services, reste une bête machine qui ne fait que ce qu'on veut bien lui dire. Pas de prise d'initiative, juste des instructions suivies à la lettre.

Ainsi, si votre page web est réalisée en français, il faut le lui dire. Si vous lui indiquez qu'elle est en anglais, elle lira les textes comme si c'était de l'anglais, même si c'est, en réalité, en français.

ACTUALITÉ French Touch : le CV anglais du président Sarkozy lu à la française

Difficile de résister au plaisir de partager avec vous cette actualité, certes dépassée, mais qui permet de se rendre compte concrètement de ce que peut donner une page rédigée en anglais et lue par une technologie d'assistance à la française... Les concepteurs ont déclaré la page en français, tout en écrivant le texte en anglais. Bref, la déclaration de langue du document n'est pas pertinente.

Le site d'actualité Écrans a fait un enregistrement de la page avec une synthèse vocale, que vous pourrez écouter d'une oreille amusée sur leur article dédié à ce cafouillage :

► <http://ur1.ca/a5axa>^a

Alors, comment trouvez-vous l'accent français de cette synthèse vocale ?

a. <http://www.ecrans.fr/Elysee-fr-Le-President-of-ze,9545.html>

Très concrètement, pour indiquer la langue d'une page, il suffit d'ajouter à la balise `<html>` l'attribut `lang`, suivi du code ISO de la langue en question. Pour une page en français, cela donne le code suivant :

```
<html lang="fr">
```

Dans le cas d'une page conçue en XHTML, la déclaration de la langue du document s'écrit ainsi :

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">
```

RESSOURCES Déclarer la langue et le code ISO

Retrouvez ces informations et la liste des codes ISO, en anglais, sur le site de w3schools.com :

► <http://ur1.ca/a5b0y^a>

Notons que deux codes ISO sont acceptés : les codes ISO 639-1 et ISO 639-2, la version 1 codant sur deux lettres et la version 2 sur trois lettres. Les deux valident le critère. Par exemple, `fr` (639-1) et `fra` (639-2) sont équivalents.

En savoir plus sur les codes de langue ISO sur Wikipédia :

► <http://ur1.ca/a6x7k^b>

a. http://www.w3schools.com/tags/ref_language_codes.asp

b. http://fr.wikipedia.org/wiki/ISO_639

Un titre à la page : la balise `<title>`

En plus de la langue, il est important de donner un titre à votre document. Attention, nous ne parlons pas des titres (balises `<h>`) servant à structurer la page. Le titre de la page est donné par la balise `<title>` au sein de la partie délimitée par les balises `<head>`. Les informations données dans cette partie ne sont pas restituées dans la page, mais elles sont utilisées par les agents utilisateurs et les robots.

Le titre de la page est affiché par le navigateur dans sa barre supérieure, au-dessus de la barre d'adresses, ainsi que dans les onglets.

Cela sera indispensable notamment si un internaute aveugle navigue entre plusieurs onglets, afin de se repérer plus facilement.

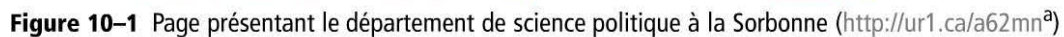
Voici un exemple d'utilisation :

```
<html>
<head>
<title>Livre accessibilité web</title>
</head>
```

Évidemment, ce titre devra être pertinent, et donc correspondre à ce dont parle la page.

Il est important de bien baliser sa page pour que les agents utilisateurs puissent récupérer l'information. Mais il est tout aussi important de ne pas « tricher » avec les balises, en les détournant de leur usage à des fins de présentation.

Prenons un exemple, avec le site de l'université de la Sorbonne.



On voit sur la capture d'écran que l'espace entre les deux titres « UFR11 - Science politique » et « Présentation générale » est simulé par une balise `<h3>` vide. Un lecteur d'écran annoncera donc un titre de niveau 3, mais sans contenu, ce qui ralentit la navigation et peut perturber l'utilisateur. On retrouve ce problème sur la page

d'exemple avec des balises `<p>` vides utilisées uniquement pour créer un effet visuel d'espacement entre des blocs de texte.

Le critère AccessiWeb 8.9 cherche à vérifier que les balises ne sont pas utilisées uniquement à des fins de présentation. Détourner une balise de son usage pour faire de la mise en page, outre le fait qu'il s'agit d'une mauvaise pratique en général, invalide donc le critère 8.9.

Notons que les balises non sémantiques `<div>` et `<span>` ne sont pas concernées, puisqu'elles ont vocation à être couplées à des classes CSS pour être mises en page.

De même, AccessiWeb autorise les tableaux de présentation. L'utilisation de la balise `<table>` à des fins de présentation est donc autorisée, à condition bien sûr que le tableau soit compréhensible quand il est linéarisé (voir la thématique 5 « Tableau », dans le chapitre 8).

À SUIVRE Séparation du fond et de la forme

Notons que le critère 8.9 vise à s'assurer d'une bonne séparation du fond (contenu/structure) et de la forme (présentation). Le critère 10.2 complète ce dispositif en demandant à utiliser les feuilles de styles pour contrôler la présentation. Voir dans la thématique « Présentation de l'information », la section « Utiliser les feuilles de styles ».

Indiquer les modifications au sein de la page

Le cadre du document est clair : on sait quelle est la grammaire utilisée, la langue et quel est le titre du document. Mais il peut arriver par endroit que des passages ne suivent pas ces déclarations. C'est le cas pour la langue et le sens de lecture.

Les changements de langue au sein de la page

Votre page est en français, mais il vous arrive de citer des passages d'un article en anglais ? Dans ce cas, il sera nécessaire d'indiquer aux technologies d'assistance que ce passage en question est en anglais. Pour ce faire, rien de plus simple : utiliser l'attribut `lang`, comme vu plus haut, sur l'élément concerné.

Prenons un exemple : le site de l'Open World Forum, en français, utilise des menus de navigation en anglais (figure 10-2, page suivante).

Si le site avait déclaré la langue de la page, elle aurait été en français, puisque le contenu principal est en français.

Figure 10-2

En-tête du site de l'Open World Forum, avec un menu de navigation utilisant des mots anglais
(<http://www.openworldforum.org>)



Mais le texte du menu est, lui, en anglais. Voyons le code source :

```
<ul>
<li class="think"><a title="Think" href="/Univers/Think"><span>THINK</span></a>
</li>
<li><a title="Innovation Awards" href="/Univers/Think/Innovation-Awards">
  >Innovation Awards</a></li>
</ul>
</li>
<li class="code"><a title="Code" href="/Univers/Code"><span>CODE</span></a>
</li>
<li class="experiment"><a title="Experiment" href="/Univers/Experiment"><span>
  >EXPERIMENT</span></a>
</li>
</ul>
```

La question de la langue doit être étudiée *du point de vue d'une synthèse vocale*. Ainsi, si un mot étranger a une prononciation identique dans la langue principale, inutile de lui appliquer un changement de langue. C'est le cas ici du mot « code » qui est sans doute en anglais, comme le reste, mais qui se prononce exactement comme en français. Par contre, le mot anglais « *think* » sera prononcé « tinqué » si aucun changement de langue n'est appliqué, ce qui est incompréhensible.

C'est donc bien la *prononciation* qui est à étudier.

Voyons comment corriger ce code pour le rendre accessible :

```
<ul>
<li class="think"><a title="Think" href="/Univers/Think"
lang="en"><span>THINK</span></a>
<ul>
<li><a title="Innovation Awards" href="/Univers/Think/Innovation-Awards"
lang="en">Innovation Awards</a></li>
</ul>
</li>
<li class="code"><a title="Code" href="/Univers/Code"><span >CODE</span></a>
</li>
<li class="experiment"><a title="Experiment" href="/Univers/Experiment"
lang="en"><span >EXPERIMENT</span></a>
</li>
</ul>
```

Dans le cas présent, nous aurions pu indifféremment appliquer l'attribut `lang` sur la balise `<li>` ou même la balise `<span>`, voire sur la balise `<ul>` une fois pour toutes.

Notons que les utilisateurs de synthèse vocale peuvent ajouter des mots au dictionnaire du logiciel et leur indiquer comment les prononcer. Ainsi, le mot « Facebook » qui, si on ne lui applique pas l'attribut `lang="en"`, est prononcé « fasseboque », peut être intégré au logiciel, auquel on indique la prononciation « fêcebouque ».

Les mots à prononcer en anglais sont fréquents, que ce soit pour des noms propres comme « Google » ou « Facebook », ou des noms communs comme « newsletter ».

Heureusement, des cas particuliers sont prévus par AccessiWeb (<http://ur1.ca/a6aru>²) :

- Si c'est un nom propre, le critère est non applicable.
- S'il s'agit d'un nom commun de langue étrangère présent dans le dictionnaire officiel de la langue par défaut de la page web (pour la France, il s'agit du dictionnaire de l'Académie française³), le critère est non applicable.
- Si le terme de langue étrangère est soumis via un champ de formulaire et réaffiché dans la page (par exemple, comme indication du terme recherché dans le cas d'un moteur de recherche), le critère est non applicable.
- Si c'est un nom commun de langue étrangère, non présent dans le dictionnaire officiel de la langue par défaut de la page web mais passé dans le langage commun de cette langue (exemple : « newsletter »), le critère est applicable, uniquement lorsque l'absence d'indication de langue peut provoquer une incompréhension pour la restitution.

2. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#cpCrit8-7

3. <http://www.academie-francaise.fr/dictionnaire/index.html>

- Si on ne peut déterminer la langue d'un passage de texte, le critère est non applicable.

Malgré ces cas particuliers, on note que même s'il est techniquement très facile d'indiquer un changement de langue, il est également *très facile d'invalider le critère 8.7*. Car même si une fonction dans l'éditeur de texte permet aux contributeurs de faire la manipulation simplement, sans taper du code, encore faut-il y penser !

ASTUCE Les abréviations, avec intitulé et <title> de langues différentes

Comment faire dans le cas d'une abréviation dont la prononciation est en français, mais l'infobulle en anglais ? Par exemple, pour PDF ? La réponse avec du code :

```
<p><abbr title="Postscript Document Format"
lang="en"><span lang="fr">PDF</span></abbr></p>
```

Les changements de sens de lecture

Si vous faites votre site en hébreu ou en arabe, par exemple, vous aurez besoin d'indiquer que le sens de lecture est de droite à gauche. De même si vous citez un article en arabe ou en hébreu dans une page en français.

Pour ce faire, il vous suffit d'utiliser l'attribut `dir`. Cet attribut spécifie une direction. Il peut prendre deux valeurs :

- de gauche à droite (`ltr` pour l'anglais *left-to-right*) – c'est la valeur par défaut ;
- de droite à gauche (`rtl` pour l'anglais *right-to-left*).

Par exemple, quand on écrit : « L'arabe (العربية, *al arabya*) est une langue originaire de la péninsule Arabique », on devrait coder de cette façon :

```
<p>L'arabe (<span lang="ar" dir="rtl">العربية</span>, <i>al arabya</i>)
est une langue originaire de la péninsule Arabique</p>
```

Ici, les changements de langue et de direction ont été correctement indiqués sur le site dont la citation est extraite : Wikipédia⁴.

RGAA Pas de test sur le sens de lecture

Notons un oubli du RGAA dans sa version 2.2.1, qui ne prévoit pas de test pour le sens de lecture, contrairement aux WCAG 2.

4. <http://fr.wikipedia.org/wiki/Arabe>

Nous venons de dresser la liste des critères de la thématique « Éléments obligatoires ». Mais pour que les technologies d'assistance puissent interpréter le code, il ne suffit pas que ces éléments soient présents, encore faut-il que l'information de la page soit structurée.

Structuration de l'information

« Utiliser des titres, des listes, des abréviations et des citations pour structurer l'information. »

<http://ur1.ca/a0cev>⁵

Nous avons, en début d'ouvrage, comparé un site web à un bâtiment ; poursuivons notre analogie. Le document HTML représente la charpente. Les feuilles de styles CSS seraient, quant à elles, la décoration. Il ne suffit pas de dessiner une porte sur un mur pour qu'une nouvelle pièce soit créée. De même, pour créer une nouvelle section d'un document, il ne suffit pas de mettre en valeur visuellement un titre via CSS pour en faire un titre. C'est dans le document HTML que la structuration se joue.

La structuration de l'information est donc primordiale pour donner corps aux différents éléments de la page, les structurer de façon cohérente afin de donner du sens.

Les premiers bénéficiaires seront les utilisateurs de lecteurs d'écran, qui vont pouvoir se représenter le plan de la page grâce aux différents niveaux de titre, ou afficher le nombre d'éléments dans une liste avant de commencer à la parcourir. Mais les personnes ayant un handicap moteur en profitent également, puisque cela leur permet de naviguer de titre en titre, par exemple.

« Avant donc que d'écrire, apprenez à penser » écrivait Nicolas Boileau. Cette citation pourrait servir de fil conducteur pour cette thématique : la pensée avant l'écriture, la structure et la sémantique avant la mise en page. Voyons de quoi il retourne précisément.

INFO Les critères de la thématique « Structuration de l'information »

La thématique « Structuration de l'information » contient cinq critères, dont trois de niveau Bronze et deux de niveau Or.

Nous traiterons de tous les critères présents dans la thématique.

5. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#structure

Des titres pour structurer la page

Toute page web doit avoir au minimum un titre de niveau 1 (balise `<h1>`). Mais selon la composition de la page, il n'est pas rare que la page contienne plus d'un titre.

L'essentiel dans ce cas est de respecter la hiérarchie des titres. On ne saurait passer d'un `<h2>` à un `<h5>` sans être passé avant par un `<h3>` et un `<h4>`. L'important est de composer une structure logique de titres représentant la composition de la page. C'est un peu comme si vous deviez faire un plan de page en reprenant la liste des titres. Tous les titres qui sont identifiés comme tels visuellement doivent être codés avec une balise `<h>` appropriée.

Prenons un exemple : le site de la Banque populaire représenté sur la figure 10-3. Nous voyons que visuellement, cette page est structurée. Elle devrait comporter plusieurs titres pour identifier chacune des grandes sections de la page.

Figure 10-3

Site de la Banque populaire
rives de Paris
(<http://www.rivesparis.banquepopulaire.fr>)



Zoomons maintenant sur une portion de la page seulement, que nous allons étudier en détail (figure 10-4).

10 – Éléments obligatoires, structuration de l'information et présentation de l'information

Figure 10–4
Zoom sur la page
de la Banque populaire
vue dans la capture
précédente.



La fenêtre sur la gauche de la capture représente l'analyse effectuée par l'extension Firefox HeadingsMap. Cette extension recense tous les titres de la page, ainsi que leur niveau, et indique en rouge les problèmes de hiérarchie des titres.

OUTILS Afficher la liste des titres d'une page avec HeadingsMap

Cette extension de Firefox est très pratique pour détecter en un clic les problèmes de hiérarchie des titres. Elle ne dira rien, en revanche, des titres manquants et il vous faudra donc compléter cette analyse par un examen humain.

Pour télécharger l'extension HeadingsMap :

► <http://ur1.ca/a0qgb>^a

a. <https://addons.mozilla.org/fr/firefox/addon/headingsmap/>

Voici le schéma que représente HeadingsMap :

- 2-Actualités
 - 3-Protégez votre famille
 - 3-Le « sans contact » débarque
 - 5-Ventes Flash
 - 4-Visitez Paris à 150 m d'altitude
- 2-Ma banque en pratique
 - 4-Accueil en langue des signes française
 - 4-La Fondation Banque populaire rives de Paris
 - 4-Communiqué de presse
 - 4-Votre conseiller en direct
 - 4-Les incontournables

On voit ici, par exemple, que la page n'a pas de niveau 1 et commence tout de suite par un `<h2>`. Par ailleurs, les titres « Ventes Flash » (en `<h5>`) et « Visitez Paris à 150 m d'altitude » (`<h4>`) sont visuellement de même niveau que le titre de niveau 2 « Actualités ». Or, codés comme ils le sont, ils semblent découler du titre `<h3>` précédent « Autorisations de découvert », alors que les informations données n'ont rien à voir avec l'article précédent.

Par ailleurs, on voit que le bloc titré en `<h2>` « Ma banque en pratique » regroupe cinq articles ayant des titres de niveau 4, alors qu'il n'y a pas de `<h3>` intermédiaire.

Une solution aurait été de donner un titre de niveau 1, par exemple sur le logo, en guise de titre de la page « Banque populaire rives de Paris Banque et assurance », puis de titrer comme suit :

- 1-Banque populaire rives de Paris Banque et assurance
 - 2-Actualités
 - 3-Protégez votre famille
 - 3-Le « sans contact » débarque
 - 2-Ventes Flash
 - 2-Visitez Paris à 150 m d'altitude
 - 2-Ma banque en pratique
 - 3-Accueil en langue des signes française
 - 3-La Fondation Banque populaire rives de Paris
 - 3-Communiqué de presse
 - 3-Votre conseiller e direct
 - 3-Les incontournables

Il aurait également été possible de ne donner un titre `<h1>` qu'aux titres des articles, ce qui décale tout d'un niveau, comme suit :

- 1-Actualités
 - 2-Protégez votre famille
 - 2-Le « sans contact » débarque
- 1-Ventes Flash
- 1-Visitez Paris à 150 m d'altitude
- 1-Ma banque en pratique
 - 2-Accueil en langue des signes française
 - 2-La Fondation Banque populaire rives de Paris
 - 2-Communiqué de presse
 - 2-Votre conseiller en direct
 - 2-Les incontournables

À SUIVRE La hiérarchie des titres bouleversée en HTML5

Pour en savoir plus sur les modifications profondes apportées par la nouvelle façon de coder les titres en HTML5, reportez-vous au chapitre 12, « HTML5, ARIA et accessibilité ».

Des listes correctement balisées

Outre les titres, les pages sont également structurées par des listes, c'est-à-dire des énumérations d'éléments. Il y a les listes à puces (`<ul>`), les listes numérotées (`<ol>`) et les listes de définition, pour des glossaires par exemple (`<dl>`).

Le fait de pouvoir caractériser une liste permet aux technologies d'assistance d'exploiter des informations telles que la nature du bloc (c'est une liste), le type de liste (`<ul>`, `<ol>` ou `<dl>`) et le nombre d'éléments (`<li>`) qu'elle contient. Le plus important est que des fonctionnalités de navigation rapide soient affectées à l'utilisation des formats de liste : par exemple, la possibilité de parcourir rapidement les éléments de liste via les touches de direction du clavier. Au-delà de la simple indication des niveaux et du nombre d'éléments, ce sont bien ces fonctionnalités qui nous intéressent.

Ainsi, les tirets faits à la main ou les sauts de ligne pour simuler visuellement une liste ne seront pas interprétés comme une liste si les bonnes balises ne sont pas utilisées.

Étudions par exemple la liste des numéros d'urgence sur le site du Conseil général de l'Aveyron (figure 10-5, page suivante).

Voyons le code source de la section de la page concernant la liste des numéros d'urgence pour les transports :

```
<h3>Les transports</h3>
<p>
SCNF info et résa : 36 35<br>
Info état des routes : 0892 68 23 12 ( voir notre rubrique info route)
<br>
Transports à la demande et lignes régulières :
</p>
<p>
05 65 59 34 80
</p>
<p>
Aéroport de Rodez : 05 65 42 20 30
</p>
```

Voici comment cette liste aurait dû être codée pour être restituée en tant que liste :

```
<h3>Les transports</h3>
<ul>
<li>SCNF info et résa : 36 35</li>
```

Info état des routes : 0892 68 23 12 (voir notre rubrique info route)
Transports à la demande et lignes régulières : 05 65 59 34 80
Aéroport de Rodez : 05 65 42 20 30

Figure 10-5

Liste des numéros d'urgence sur le site du Conseil général de l'Aveyron
(<http://www.cg12.fr>)



Autre exemple, celui du site de la ville de Blinic, encore plus frappant puisque la liste a été simulée visuellement, mais sans utiliser les balises HTML nécessaires (figure 10-6).

Figure 10-6

Article sur les journées du patrimoine sur le site de la ville de Blinic
(<http://ur1.ca/a61y2a>)



a. http://www.ville-binic.fr/index.php?id_site=5&id_page=135&option=actualite&action=afficher_contenu&id_actualite=87

201

```
<li>Exposition en mairie.</li>
<li>Visite gratuite du musée.</li>
<li>A l'église, présentation d'une vidéo retraçant l'évolution de la pratique
religieuse depuis 1939, exposition d'ex-voto marins et de documents relatifs
aux rituels religieux dans les années 1945-1975.</li>
<li>A la Chapelle Saint-Gilles, Ville-Jacob, exposition de cartes postales et
présentation d'une vidéo sur sa restauration.</li>
<li>Initiation à la navigation sur les bateaux traditionnels : pirogue baleinière,
doris, vieux gréement ; « Le Rigel » de 17h à 19h (sur réservation)</li>
<li>Ballades au départ de l'office de tourisme :
<!-- Insertion d'une liste dans une liste, balisée à nouveau avec ul et li -->
  <ul>
    <li>De 10h à 11h15, Balade guidée « Pas à pas, pierre à pierre »</li>
    <li>De 15h à 16h, Balade sensorielle « Des Goélettes à la plaisance »</li>
  </ul>
</li>
</ul>
```

Une fois de plus, il n'y a rien de bien compliqué d'un point de vue technique. Il s'agit d'une application toute simple du HTML de base. Or, on voit dans l'exemple ci-avant que le site lui-même n'a pas été réalisé en utilisant les balises appropriées. Cela montre une mauvaise utilisation du HTML de base par les développeurs eux-mêmes.

Mais ce qui complique encore la tâche, c'est que des personnes non techniques peuvent être amenées à faire des listes. Nous pensons bien sûr aux contributeurs. Formés pour publier du contenu intéressant et utile, ils ne sont pas toujours au fait du HTML et encore moins de l'accessibilité. Seule une formation spécifique dédiée aux personnes mettant à jour les contenus peut permettre de maintenir les pages web accessibles à tous.

RGAA Des listes marquées visuellement seulement

Le RGAA et AccessiWeb n'adoptent pas le même point de vue au sujet des listes. Alors que le RGAA restreint la notion de liste à la présence « visuelle » d'un marqueur de liste, AccessiWeb considère qu'il s'agit d'une erreur d'interprétation des WCAG. Par exemple, pour le RGAA, un menu de navigation (qui ne possède naturellement pas de marqueur « visuel ») peut très bien être codé en une suite indéterminée de balises `<a>` avec toute la complexité en restitution et le manque d'utilisabilité que cela implique pour l'utilisateur.

Des abréviations et acronymes explicites

N'avez-vous jamais atterri sur une page web dont vous arriviez à lire le texte, mais dont le sens vous échappait partiellement en raison de l'emploi de nombreux acronymes ou abréviations que vous ne connaissiez pas ? On se sent bête, n'est-ce pas ? On peut toujours faire une recherche pour savoir de quoi il s'agit, mais il est plus vraisemblable que l'on quitte la page.

Les critères 9.3 (présence) et 9.4 (pertinence) visent à prévenir cette situation. Ils sont tous deux de niveau Or, ce qui les rend non obligatoires, mais leur utilité et leur facilité d'implémentation nous ont poussé à vous en parler.

Distinguons deux types de textes abrégés :

- les *abréviations*, qui se prononcent lettre par lettre (sigles), comme HTML, RGAA, WCAG, W3C.
- les *acronymes*, qui se prononcent comme des mots : ONU, Onisep, Unesco...⁶

Quand il s'agit d'une abréviation, il est possible de mettre une balise spécifique `<abbr>` pour indiquer que le texte doit se prononcer lettre par lettre. Ce n'est pas utile pour un acronyme qui se lit comme un mot. Bien qu'il ne s'agisse pas d'une règle d'accessibilité et qu'il soit difficile en pratique de baliser tous les acronymes et abréviations, la balise `<abbr>` peut être utilisée à cette fin.

À SAVOIR L'attribut acronym est obsolète en HTML5

Notons que l'attribut `acronym`, dédié aux acronymes, est rendu obsolète en HTML5. Il n'est donc plus recommandé de l'utiliser. Il ne reste plus que l'attribut `abbr`, utilisable sur les abréviations comme sur les acronymes.

Cette simplification est judicieuse, car il était parfois difficile pour certains de faire la différence entre abréviation et acronyme. Par ailleurs, un acronyme ne posant pas de difficulté de lecture, rendre son balisage facultatif est une bonne chose. Reste bien sûr à définir abréviations et acronymes, et nous allons voir que plusieurs techniques sont permises.

Plusieurs solutions s'offrent à vous pour définir une abréviation ou un acronyme.

Vous pouvez le faire, comme dans ce livre, en clair dans le texte de l'article, entre parenthèses par exemple.

`<p>Le référentiel AccessiWeb et le <abbr>RGAA</abbr> (référentiel général d'accessibilité pour les administrations) permettent tous deux une vérification de la conformité aux <abbr>WCAG</abbr> (règles d'accessibilité des contenus web) du W3C (<span lang="en">World Wide Web Consortium</span>)</p>`

Un texte en clair est plus facilement lisible, car tout le monde peut le voir instantanément. Mais ça peut devenir un peu lourd quand les abréviations ou acronymes se succèdent dans la page.

6. Il est d'ailleurs intéressant de remarquer que, en français, les acronymes de plus de trois lettres peuvent être écrits en bas de casse (avec une capitale initiale), comme des mots. (Voir le *Lexique des règles typographiques en usage à l'imprimerie nationale*, 2002, page 160... et l'usage de notre éditeur !)

Une autre technique consiste à indiquer la définition au sein d'un attribut `title`.

```
<p>Le référentiel AccessiWeb et le <abbr title="référentiel général  
d'accessibilité pour les administrations">RGAA</abbr> permettent tous deux une  
vérification de la conformité aux <abbr title="règles d'accessibilité pour les  
contenus web">WCAG</abbr> du <abbr title="World Wide Web Consortium"  
lang="en"><span lang="fr">W3C</span></abbr> </p>
```

Enfin, vous pouvez prévoir un glossaire reprenant tous les acronymes et abréviations de votre site, disponible par exemple en pied de page, et faire un lien vers ce glossaire à chaque fois qu'un acronyme ou une abréviation apparaît dans la page.

Notons que seule la première occurrence nécessite d'être définie. Mais nous parlons bien sûr de la première occurrence du terme dans la page web, et non dans le site. Avec les moteurs de recherche, chaque page web peut être la porte d'entrée sur votre site, ne l'oubliez pas.

Des citations déclarées comme telles

Les citations, si elles sont correctement déclarées, peuvent fournir des informations précieuses, telles que la source de la citation. Malheureusement, les lecteurs d'écran ne prennent pas en compte les citations, ce qui est fort dommage car cela permettrait une restitution particulière par la synthèse vocale, avec une voix différente pour indiquer qu'il s'agit d'une citation, par exemple.

Bien que cette règle soit mal implémentée, déclarer une citation est une règle d'accessibilité de niveau Bronze.

DÉBAT Baliser les citations, inutile pour l'accessibilité ?

En pratique, le balisage des citations n'apporte rien aux utilisateurs en situation de handicap. Il s'agit d'une bonne pratique de codage, certes, mais comme aucune fonctionnalité n'a été associée par les technologies d'assistance, le bénéfice de ce balisage est nul. En faire une règle de niveau Bronze prête donc à débat...

Les balises concernées pour indiquer une citation sont les balises `<q>`, indiquant une citation en ligne, et `<blockquote>` pour une citation longue, sous forme de bloc. Ces deux balises peuvent recevoir l'attribut `cite` pour indiquer l'URL menant vers le document dont est extraite la citation.

Si nous reprenons la citation en introduction de thématique, le code serait :

```
<p><q>« Avant donc que d'écrire, apprenez à penser »</q> écrivait Nicolas Boileau.</p>
```

On pourrait indiquer la source de cette citation, avec l'attribut `cite`, de cette façon :

```
<p><q cite="http://www.poesie-francaise.fr/nicolas-boileau/poeme-l-art-poetique-chant-I.php">« Avant donc que d'écrire, apprenez à penser »</q> écrivait Nicolas Boileau.</p>
```

Dans le cas d'une citation plus longue, on indiquerait un bloc de citation de cette façon :

```
<p>Nicolas Boileau écrivait, dans son texte De l'Art poétique (Chant 1)&nbsp; ;</p>
<blockquote cite="http://www.poesie-francaise.fr/nicolas-boileau/poeme-l-art-poetique-chant-I.php">
Ce que l'on conçoit bien s'énonce clairement,<br />
Et les mots pour le dire arrivent aisément.
</blockquote>
```

ATTENTION La balise `<cite>` ne sert pas à identifier une citation !

Nous avons vu l'attribut `cite=""`. Il existe aussi la balise `<cite>`, qui ne sert pas à entourer une citation, mais à identifier une référence, comme un livre.

Si nous reprenons l'exemple précédent, voici comment nous pouvons utiliser la balise `<cite>` en complément de la balise `<blockquote>`.

```
<p>Nicolas Boileau écrivait, dans son texte <cite>De l'Art poétique (Chant 1)</cite>&nbsp; ;</p>
<blockquote cite="http://www.poesie-francaise.fr/nicolas-boileau/poeme-l-art-poetique-chant-I.php">
Ce que l'on conçoit bien s'énonce clairement,<br />
Et les mots pour le dire arrivent aisément.
</blockquote>
```

Pour en savoir plus sur cette balise `<cite>`, dont l'utilisation varie quelque peu en HTML5 (contrairement à ce qu'on trouvait dans HTML4, les noms d'auteur ne sont plus encadrés par `<cite>`), voir le site du W3C :

► <http://ur1.ca/a8ca0a>

a. <http://dev.w3.org/html5/spec/single-page.html#the-cite-element>

Prenons un autre exemple, tiré cette fois du blog de la mairie de Trévoux.

Figure 10-7

Article du blog de la mairie de Trévoux, citant le préfet (<http://mairie-trevoux.over-blog.com>)



Dans cet exemple, on peut lire une citation longue du préfet, codée de cette façon :

```
<p style="margin: 0cm 0cm 10pt;">
<span style="font-size: 12pt;"><span style="font-family: Calibri;">Monsieur le
Préfet nous répond&nbsp;  : «&nbsp;  ...Toutefois, les évènements tels que les tempêtes, la
grêle, le poids de la neige sur les toitures ou le gel relèvent des garanties
contractuelles conclues entre l'assuré et son assureur. Pour cette raison, les
dommages causés par de tels événements ne peuvent être couverts au titre de la
reconnaissance de l'état de catastrophe naturelle. Au vu de ce qui précède, et malgré
l'intensité de cet orage, je suis au regret de vous informer que je ne peux pas
donner suite à votre demande.&nbsp;  »</span></span>
</p>
```

Voici comment déclarer ce bloc de citation de façon accessible :

```
<p>Monsieur le Préfet nous répond&nbsp;  </p>
<blockquote>
«&nbsp;  ...Toutefois, les évènements tels que les tempêtes, la grêle, le poids de la
neige sur les toitures ou le gel relèvent des garanties contractuelles conclues entre
l'assuré et son assureur. Pour cette raison, les dommages causés par de tels
événements ne peuvent être couverts au titre de la reconnaissance de l'état de
catastrophe naturelle. Au vu de ce qui précède, et malgré l'intensité de cet orage,
je suis au regret de vous informer que je ne peux pas donner suite à votre demande.
</blockquote>
```

Dans les feuilles de styles par défaut des navigateurs récents, l'utilisation de la balise `<q>` génère des guillemets en CSS, tandis que la balise `<blockquote>` provoque un retrait du texte des deux côtés.

Mais comme nous l'avons vu, ce qui doit présider au choix d'utiliser ou non ces balises, c'est la nature de l'élément. S'agit-il effectivement d'une citation ? Si oui, il faut utiliser la balise appropriée, quitte à modifier l'apparence par défaut avec des feuilles de styles. Sinon, il ne faut pas les utiliser.

Une fois la structure mise en place, il sera temps de s'occuper de la présentation de l'information et de son accessibilité.

Présentation de l'information

« Utiliser des feuilles de styles pour contrôler la présentation de l'information. Vérifier l'effet de l'agrandissement des tailles de caractères sur la lisibilité. S'assurer que les liens sont correctement identifiables, que la prise de focus est signalée, que l'interlignage est suffisant et donner la possibilité à l'utilisateur de contrôler la justification des textes. S'assurer que les textes cachés sont correctement restitués et que l'information n'est pas donnée uniquement par la forme ou la position d'un élément. »

<http://ur1.ca/a0cf4>⁷

Ça y est ! Vous avez la structure de votre site, vos contenus sont prêts à être intégrés. On va enfin pouvoir passer à la partie la plus amusante : la mise en page !

Nous vous mettions en garde précédemment sur le mépris que représentait une version accessible moche par rapport à un site original soigné (voir l'encart dans le chapitre 6, page 115). Il est donc important de soigner, pour tous vos utilisateurs, la présentation de vos contenus. Rappelons que l'accessibilité web vise toutes les situations de handicap, et pas seulement les utilisateurs aveugles.

La présentation de l'information concerne d'ailleurs aussi bien les personnes malvoyantes ou aveugles que les personnes ayant un handicap moteur, ou encore un handicap cognitif comme la dyslexie.

7. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#presentation

INFO Les critères de la thématique « Présentation de l'information »

La thématique « Présentation de l'information » contient quinze critères, dont huit de niveau Bronze, deux de niveau Argent et cinq de niveau Or.

Nous ne traiterons pas les critères 10.8, 10.10 et 10.11, tous trois de niveau Or et ne relevant vraiment que d'une question d'optimisation du confort d'utilisation.

Retrouvez ces critères sur le référentiel AccessiWeb.

Critère 10.8 (choix des couleurs par l'utilisateur) :

▶ <http://ur1.ca/a4j9d^a>

Critère 10.10 (pas de barre de défilement horizontal) :

▶ <http://ur1.ca/a4j9g^b>

Critère 10.11 (des blocs de moins de 80 caractères) :

▶ <http://ur1.ca/a4j9s^c>

- a. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-10-8
- b. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-10-10
- c. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-10-11

Utiliser les feuilles de styles

Le premier critère de la thématique revient sur la notion de séparation du fond (contenus/structure) et de la forme (présentation) en demandant à ce que la page utilise des feuilles de styles pour contrôler la présentation de l'information.

Une liste de balises et attributs servant à la présentation sont interdits.

Tableau 10-1 Balises et attributs interdits

Balises	Attributs	Balises	Attributs
<basefont>	align	<u>	text
<blink>	alink		vlink
<center>	background		width (sauf sur la balise où width est autorisé)
	basefont		height (sauf sur la balise où height est autorisé)
<marquee>	bgcolor		
<s>	border		
<strike>	color		
<tt>	link		

Outre l'interdiction d'utiliser certains éléments et attributs, le critère 10.1 cherche à vérifier que les espaces ne sont pas utilisées pour faire de la mise en page, comme pour simuler un tableau, par exemple.

À SAVOIR Les balises typographiques `<b>` et `<i>`

Vous remarquerez que les balises `<b>` et `<i>` ne sont pas interdites. Certains pensent, à tort, qu'il faut systématiquement les remplacer par les balises `<strong>` et `<em>`. Il est vrai que le rendu par défaut dans les navigateurs est le même : `<b>` et `<strong>` produisent un texte en gras, tandis que `<i>` et `<em>` produisent un texte en italique. Mais alors que `<strong>` indique un texte important, `<b>` est utilisé pour mettre en valeur certains éléments, comme des mots-clés ou le chapeau d'un article, par exemple. De même, `<em>` sert à indiquer un mot ou un groupe de mots sur lequel on veut appuyer, tandis que `<i>` s'applique aux textes que l'on souhaite voir prononcés avec une voix ou une attitude différente. `<i>` est par exemple utilisée pour indiquer qu'il s'agit d'un terme technique, d'un texte dans une langue différente, d'une pensée ou encore d'un rêve.

Pour en savoir plus sur chacun de ces éléments, vous pouvez consulter la section dédiée aux éléments de texte sémantiques dans la spécification de HTML5 (<http://ur1.ca/a64ye^a>), et plus précisément :

- `<strong>` : <http://ur1.ca/a64yu^b>
- `<b>` : <http://ur1.ca/a64zo^c>
- `<em>` : <http://ur1.ca/a64zz^d>
- `<i>` : <http://ur1.ca/a650f^e>

- a. <http://dev.w3.org/html5/spec/single-page.html#text-level-semantics>
- b. <http://dev.w3.org/html5/spec/single-page.html#the-strong-element>
- c. <http://dev.w3.org/html5/spec/single-page.html#the-b-element>
- d. <http://dev.w3.org/html5/spec/single-page.html#the-em-element>
- e. <http://dev.w3.org/html5/spec/single-page.html#the-i-element>

OUTILS Tester l'absence de balises et d'attributs interdits avec Tanaguru

Pour vérifier à la main qu'une page n'utilise pas de balises et d'attributs interdits, il est possible d'afficher le code source et d'utiliser le raccourci clavier `Ctrl+F` pour rechercher, un par un, les balises et attributs pour s'assurer qu'ils sont bien absents de la page.

Vous pouvez donc faire cette démarche longue et fastidieuse, ou alors utiliser Tanaguru. Cet outil en ligne de test automatique de l'accessibilité évalue également d'autres aspects. Mais dans le cas du critère 10.1, il est particulièrement utile.

Utiliser Tanaguru en ligne :

- <http://my.tanaguru.com>

Notons qu'une fois la procédure effectuée, il vous restera à vérifier que les espaces (` `) ne sont pas utilisées pour faire de la mise en page.

À ce titre, le critère 10.1 est le pendant du critère 8.9 interdisant les détournements de balises à des fins de présentation (voir dans la première thématique de ce chapitre, la section « Ne pas induire les agents utilisateurs en erreur »). Le critère 10.1 complète ce dispositif quand la mise en page est simulée en détournant les espaces, qui ne sont pas des balises, mais dont l'objet n'est quand même pas la mise en page.

À SAVOIR La balise et l'attribut style

Notons que l'usage de la balise ou de l'attribut `style` est autorisé, bien que ceux-ci apportent dans la page HTML ou sur un élément particulier des indications de mise en forme en CSS. Toutefois, la spécification précise clairement que tous deux sont déconseillés en production. Ils sont malgré tout autorisés car ils restent pratiques dans le cadre d'un prototypage.

Rendre l'information perceptible

Vous avez pensé à bien structurer votre page, en évitant de mélanger les contenus avec la présentation. Pourtant, il reste encore à s'assurer que la présentation de ces contenus est bien perceptible, que ce soit en désactivant les images ou les CSS, en effectuant un zoom texte ou en restituant des textes cachés si nécessaire.

Désactiver les images ou les CSS

L'une des règles consiste à permettre aux personnes ayant des difficultés à percevoir les contenus visuels d'accéder aux contenus d'une autre façon. Il est nécessaire pour cela que le contenu ne dépendent pas de la présence des images ou de la mise en page.

Les images sont traitées par la thématique 1 (voir le chapitre 6). L'objectif de la désactivation des images ici est plutôt de vérifier que celle-ci n'entraîne pas une modification de la mise en page qui ferait disparaître des informations. De même pour la désactivation des CSS. Notons qu'il s'agit de désactiver l'un *ou* l'autre, pas de faire un test avec images et CSS désactivées en même temps !

OUTILS Désactiver images et CSS avec la Web Developer Toolbar

Pour désactiver les images, rendez-vous dans le menu *Images* de la barre d'outils Web Developer Toolbar et cliquez sur *Désactiver les images > Toutes les images*.

Pour désactiver les CSS, réaffichez d'abord les images si nécessaire, puis dans le menu *CSS*, cliquez sur *Désactiver les styles CSS > Tous les styles*.

Le mieux est encore de prendre un exemple concret. Voici une page du site de la ville d'Ermont (figure 10-8). On y voit notamment un menu de navigation en haut de page, des logos des réseaux sociaux et des images avec une loupe pour agrandir ou réduire la taille du texte et imprimer, puis encore des images-liens dans le corps de l'article pour mener aux différentes sous-rubriques.

10 – Éléments obligatoires, structuration de l'information et présentation de l'information

Figure 10-8

Page de la ville d'Ermont consacrée aux services municipaux, avec images et CSS actives ([http://ur1.ca/a66bp^a\)](http://ur1.ca/a66bp^a))

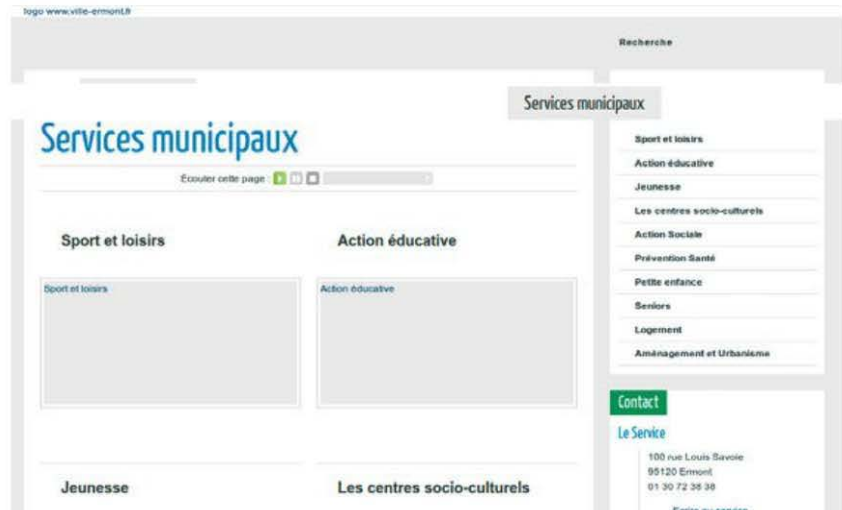


a. <http://www.ville-ermont.fr/services-municipaux/>

Maintenant, désactivons les images (figure 10-9).

Figure 10-9

Même page que sur la capture d'écran précédente, mais avec les images désactivées



Et là, c'est le drame ! On perd le menu de navigation principal, ainsi que les liens de partage vers les réseaux sociaux et ceux pour agrandir/réduire la taille du texte ou imprimer. Par contre, on voit que l'alternative a bien été renseignée sur les images indiquant les sous-rubriques, le texte de l'alternative apparaissant bien à la place de l'image. Alors, que s'est-il passé ?

Concernant le menu de navigation, c'est pourtant du texte. Mais c'est un texte en blanc avec une image de fond noire qui, une fois désactivée, rend invisible le texte qui devient blanc sur fond blanc (nous verrons en détail ce problème dans la section qui suit).

Concernant les réseaux sociaux, voici le code HTML :

```
<div class="left">
<a id="twitter" target="_blank" title="En parler sur twitter"
href="https://twitter.com/share?url=http%3A%2F%2Fwww.ville-
ermont.fr%2Fservices-municipaux%2F%3Fno_cache%3D1">Twitter</a>
<a id="facebook" target="_blank" title="Partager sur facebook"
href="http://www.facebook.com/share.php?u=http%3A%2F%2Fwww.ville-
ermont.fr%2Fservices-municipaux%2F%3Fno_cache%3D1">Facebook</a>
<a id="share" class="modalInput" title="Envoyer à un ami" href="http://
dev.ville-ermont.fr/index.php?id=2280">Envoyer</a>
</div>
```

Étrange, penserez-vous peut-être... C'est vrai que le code HTML en lui-même ne contient pas d'image. Comment se fait-il alors que la désactivation des images rende ces textes invisibles ? La réponse se trouve dans la feuille de styles.

```
#twitter {
  background: url("../img/sprite-v.png") no-repeat scroll 0 -490px transparent;
  display: block;
  float: left;
  height: 22px;
  margin: 4px 10px 0 0;
  overflow: hidden;
  text-indent: -5000px;
  width: 20px;
}
#facebook {
  background: url("../img/sprite-v.png") no-repeat scroll -30px -490px transparent;
  display: block;
  float: left;
  height: 22px;
  margin: 4px 10px 0 0;
  overflow: hidden;
  text-indent: -5000px;
  width: 20px;
}
#share {
  background: url("../img/sprite-v.png") no-repeat scroll -60px -490px transparent;
  display: block;
  float: left;
  height: 22px;
  margin: 4px 10px 0 0;
```

```
overflow: hidden;  
text-indent: -5000px;  
width: 20px;  
}
```

Nous voyons que les images ont été insérées en CSS et que le texte a été caché en étant déporté à 5 000 pixels sur la gauche. Un lecteur d'écran pourra donc bien lire ce texte, mais pas une personne qui voit la page et qui navigue sans les images.

DÉBAT À qui ça sert, concrètement, une navigation sans images ?

Le cas d'une navigation sans images, mais avec les CSS actives, est très théorique. C'est une situation de handicap qui a été imaginée, mais qui ne touche pas précisément les personnes handicapées. Le cas d'un problème de téléchargement des images peut survenir dans certaines zones à bas débit où le temps de chargement rend l'affichage des images impossible. Mais on sort ici de l'accessibilité des personnes en situation de handicap.

Quant aux utilisateurs de lecteurs d'écran, ils naviguent sans images, certes, mais également sans CSS. La question ne se pose donc pas de la même façon.

Le référentiel AccessiWeb, à l'entrée « contenu visible » de son glossaire (<http://ur1.ca/a66o2>⁸) précise bien que cette technique n'est pas recommandée, même si elle est faite de manière accessible, ce qui n'est pas le cas ici de toutes façons :

« La pratique qui consiste à gérer des images en propriété de fond d'éléments via CSS est formellement déconseillée, même si elle est accompagnée d'un texte caché "visible" lorsque les images ou les couleurs sont désactivées. »

Réactivons les images, et voyons maintenant ce qui se passe quand on désactive les feuilles de styles (figure 10-10, page suivante).

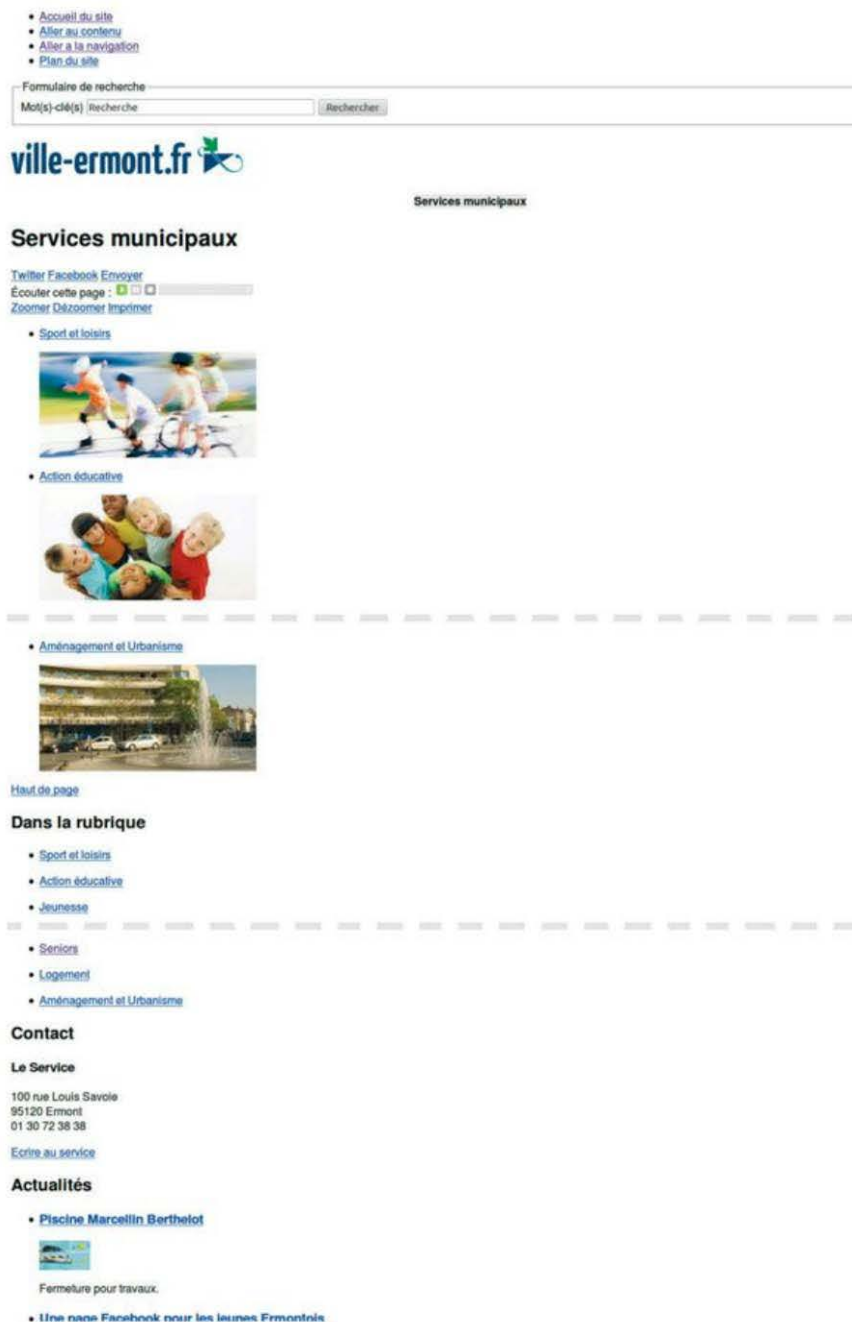
Cette fois, le menu principal apparaît, mais il n'est pas positionné en CSS (il était positionné en absolu), on le retrouve donc tout en bas de la page. Par ailleurs, on retrouve bien le texte des liens « Twitter », « Facebook » et de « Partage », puisque le texte n'est plus renvoyé hors de l'écran.

Malgré tout, le critère 10.2 est invalidé, puisque sans les images, on perd du contenu. Le critère 10.3 demande, quant à lui, à ce que les informations restent compréhensibles quand les CSS sont désactivées, ce qui est le cas pour la page étudiée.

8. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#mContVisible

Figure 10-10

Même page du site de la ville d'Ermont que précédemment, mais avec les images actives et les CSS désactivées.



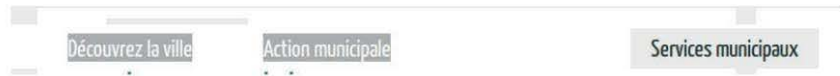
Attention aux couleurs de fond !

Revenons plus en détail sur ce menu qui disparaît quand les images sont désactivées. Nous avons vu que le problème était lié à une mauvaise gestion des couleurs de premier et d'arrière-plan. Cette question fait l'objet du critère 10.5 d'AccessiWeb, qui

demande : « Dans chaque page web, les déclarations CSS de couleurs de fond d'élément et de police sont-elles correctement utilisées ? »

Figure 10-11

Le menu de la ville d'Ermont devient blanc sur fond blanc quand les images sont désactivées.



Le texte des deux premiers menus (*Découvrez la ville* et *Action municipale*) que l'on voit apparaître sur la figure 10-11 est en réalité sélectionné avec la souris, ce qui fait qu'on peut le voir. En revanche, le troisième menu (*Ermont pratique*) est complètement invisible, car il est en blanc sur fond blanc. Seule une image de fond noire en CSS permet de rendre ces textes blancs visibles, mais quand l'image est désactivée, le texte devient invisible.

Concrètement, qu'aurait-il fallu faire ? La solution est très simple, il faut ajouter une propriété `background-color` en CSS avec une valeur proche de celle de l'image, par exemple `#000` (noire).

Cela signifie-t-il que tout texte en couleur doit avoir une couleur de fond spécifiée ? Non, bien entendu, car il ne faut pas oublier que le sigle CSS (*Cascading Style Sheet*) signifie « feuilles de styles en cascade », et que cette cascade est au fondement même du principe d'héritage des couleurs. Il vous faut donc seulement vérifier que chaque texte en couleur hérite d'au moins une couleur de fond lui assurant sa visibilité.

OUTIL Tester l'héritage des couleurs en CSS avec Firebug

Pour commencer, téléchargez et installez l'extension Firebug pour Firefox :

► <http://ur1.ca/a66ok>^a

Maintenant, repérez dans la page les textes en couleur. Inspectez l'élément avec Firebug (clic droit puis *Inspecter l'élément*).

La fenêtre de Firebug est composée par défaut du code HTML à gauche, et des CSS à droite. Sur la fenêtre des CSS, cliquez sur l'onglet *Calculé*. Toutes les propriétés CSS de l'élément inspecté s'affichent alors. Il vous suffit de vérifier qu'une couleur de fond est bien associée à l'élément, soit par la propriété `background-color`, soit par la propriété `background`.

Si aucune couleur n'est spécifiée pour l'élément testé, remontez l'héritage en cliquant, en haut de la fenêtre HTML, sur chacun des sélecteurs dont dépend l'élément, jusqu'à la balise `<html>`. Les styles CSS calculés se mettent alors à jour dans la fenêtre de droite.

Vous trouvez enfin une couleur de fond spécifiée ? Le critère est validé. Sinon, il va falloir en indiquer une, sous peine d'invalider le critère 10.5 et de voir votre texte devenir invisible dans certains cas.

a. <https://addons.mozilla.org/fr/firefox/addon/firebug/>

Le texte est-il lisible avec un zoom texte à 200 % ?

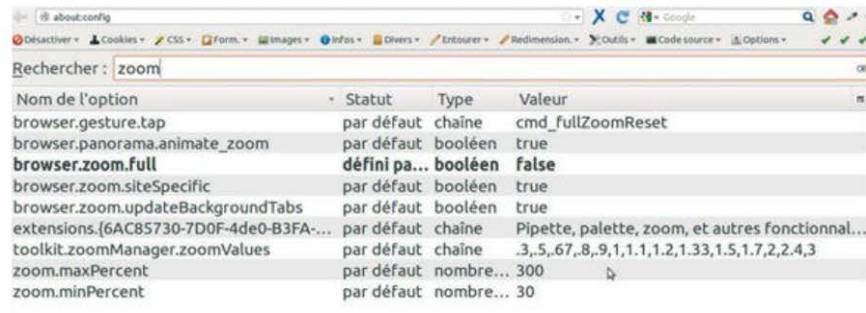
Certaines personnes ayant une vision basse peuvent avoir besoin d'agrandir la taille du texte pour le lire, sans pour autant utiliser une technologie d'assistance. Dans ce cas, c'est le zoom du navigateur qui est utilisé. Afin de ne pas déformer les images et de garder les textes bien nets, il est recommandé de n'agrandir que les textes. On fait alors un « zoom texte seulement ». Sous Firefox, rendez-vous dans le menu *Affichage > Zoom* puis cochez *Zoom texte seulement*.

Le critère 10.4 demande : « Dans chaque page web, le texte reste-t-il lisible lorsque la taille des caractères est augmentée jusqu'à 200 %, au moins ? »

Pour obtenir un zoom du texte à 200 % sous Firefox, vous pouvez vérifier les valeurs de zoom par défaut dans le panneau de configuration (que vous obtenez en tapant `about:config` dans la barre d'adresses du navigateur).

Figure 10-12

Page des valeurs de zoom dans `about:config` sous Firefox 15 pour Ubuntu



Nom de l'option	Statut	Type	Valeur
browser.gesture.tap	par défaut	chaîne	cmd_fullZoomReset
browser.panorama.animate_zoom	par défaut	booléen	true
browser.zoom.full	défini pa...	booléen	false
browser.zoom.siteSpecific	par défaut	booléen	true
browser.zoom.updateBackgroundTabs	par défaut	booléen	true
extensions.{6AC85730-7D0F-4de0-B3FA-...}	par défaut	chaîne	Pipette, palette, zoom, et autres fonctionnal...
toolkit.zoomManager.zoomValues	par défaut	chaîne	.3,.5,.67,.8,.9,1.1,1.2,1.33,1.5,1.7,2,2.4,3
zoom.maxPercent	par défaut	nombre...	300
zoom.minPercent	par défaut	nombre...	30

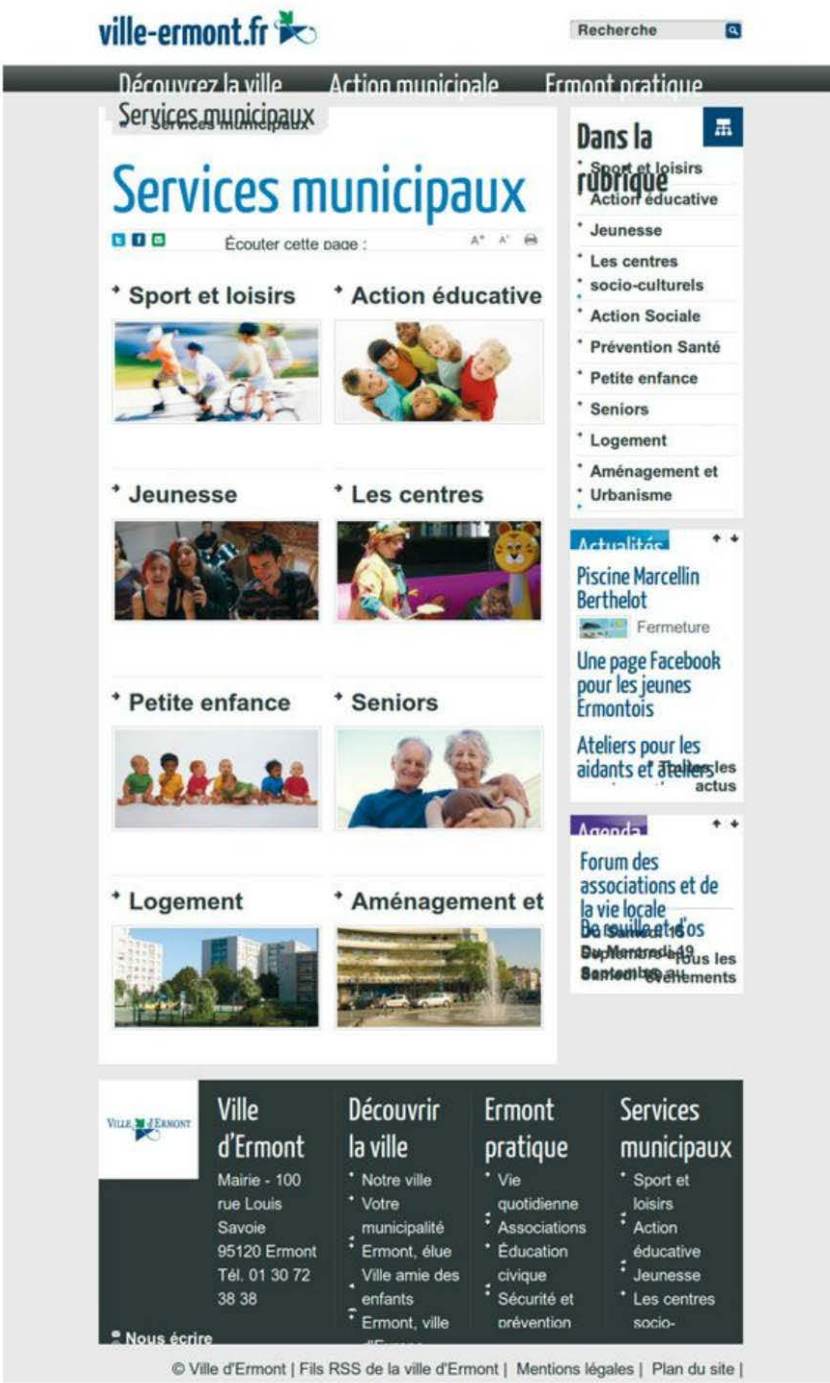
Les valeurs de zoom par défaut sous Firefox sont : 30 %, 50 %, 67 %, 80 %, 90 %, **100 %**, 110 %, 120 %, 133 %, 150 %, 170 %, 200 %, 240 %. Ainsi, si vous souhaitez augmenter le texte de 200 %, il vous suffit de vous placez sur la page à la valeur par défaut de 100 % (*Ctrl+0*) puis de zoomer en appuyant *six fois* sur les touches *Ctrl* et *+* jusqu'à obtenir 200 %.

Augmentons la taille du texte à 200 % dans notre exemple de la ville d'Ermont (figure 10-13).

La question est de savoir si le texte reste lisible. Pour cela, il faut s'assurer que les blocs ne se chevauchent pas. Dans le cas présent, c'est pourtant le cas. Le texte du menu empiète sur le contenu de la page ; les titres des blocs de droite disparaissent, car ils sont en blanc sur une image de fond colorée, et l'image n'étant pas agrandie, le texte finit par déborder sur le fond blanc de la page, devenant invisible. Enfin, plusieurs blocs de texte deviennent illisibles.

10 – Éléments obligatoires, structuration de l'information et présentation de l'information

Figure 10–13
Page de la rubrique des services municipaux de la ville d'Ermont, agrandie à 200 % (Pour des raisons de place, certaines parties de la page ont été tronquées, mais les informations apparaissent telles quelles.)



Dans l'exemple suivant, nous pouvons voir le site Service-Public.fr agrandi à 200 % (figure 10-14). Ce site est composé de nombreux blocs de texte, disposés sur pas moins de quatre colonnes. Pourtant, nous pouvons constater que le texte reste lisible quand il est agrandi.

Figure 10-14
Le site Service-Public.fr
agrandi à 200 %



Notons que le critère 10.4 proscriit l'utilisation des unités `pt`, `pc`, `mm`, `cm` et `in` pour une restitution sur écran (ces unités sont autorisées pour une sortie papier, bien sûr). De même, le pixel est proscriit pour la définition des tailles de police, comme l'indique AccessiWeb dans son glossaire à l'entrée « taille des caractères » (<http://ur1.ca/a6xjs>⁹). Pour voir la position ambiguë de la WAI à ce sujet, reportez-vous à l'encart, page 82, dans le chapitre 4.

9. http://www.accessiweb.org/index.php/glossaire_referentiel_2.2.html#mTailleCaractere

RGAA Le pixel autorisé ou pas ?

Le pixel est un cas particulier, car il n'a pas fait l'objet d'une technique suffisante, refusée par la WAI car non « compatible avec l'accessibilité ». En même temps, il n'a pas fait l'objet d'un échec (*failure*) car il est interprété correctement, comme une unité relative, par de nombreux navigateurs en dehors d'Internet Explorer. Alors qu'AccessiWeb a tranché la question en interdisant le pixel, le RGAA l'autorise pour les textes dans les éléments de formulaires.

Quoi qu'il en soit, s'il est vrai que le pixel est une unité relative, entre le fait qu'Internet Explorer l'interprète comme une unité absolue et dans la mesure où il existe d'autres méthodes (*em*, *%*), interdire le pixel pour définir la taille des textes n'est pas vraiment un problème.

Des textes cachés restitués si nécessaire

Le critère 10.13 s'intéresse aux textes cachés et à leur restitution par les technologies d'assistance.

Avant de cacher un texte, posez-vous d'abord la question de sa pertinence : cette information est-elle bien utile ? Si oui, faut-il vraiment la cacher ? En effet, une information utile à tous mérite d'être visible par tous.

Mais si vous devez vraiment cacher du texte à destination des personnes aveugles, prenez garde à le faire de façon accessible, c'est-à-dire en permettant aux technologies d'assistance de restituer l'information. Ainsi, il peut arriver que certains textes uniquement à destination de publics non-voyants soient volontairement cachés.

Revenons à notre exemple du site de la ville d'Ermont. Le formulaire de recherche est codé de cette façon :

```
<form id="search" action="http://www.ville-ermont.fr/moteur-de-recherche/"
method="get" class="artifica-form">
  <fieldset>
    <legend class="invisible">Formulaire de recherche</legend>
    <label for="query">Mot(s)-clé(s)</label>
    <input id="query" type="text" name="tx_artificasearchengine_pil[sword]"
size="30" maxlength="50" accesskey="4" tabindex="4" value="Recherche" />
    <button type="submit" class="goFind">Rechercher</button>
  </fieldset>
</form>
```

On voit qu'il a été prévu une légende (balise `<legend>`) indiquant « Formulaire de recherche » en plus de l'étiquette (balise `<label>`) « mots-clés ». En dehors d'une mauvaise utilisation de ces balises (voir la thématique 11 « Formulaires » dans le chapitre 11), on peut supposer, avec l'attribut `class="invisible"`, que cette mention vise uniquement les personnes ne voyant pas la page mais pouvant y accéder avec un lecteur d'écran.

Vérifions maintenant le style CSS associé à la classe invisible :

```
.invisible {  
    display: none !important;  
}
```

Dans ce cas précis, l'objectif de restitution par les lecteurs d'écran n'est pas atteint, car une propriété CSS `display:none`; empêche toute restitution, ce qui invalide le critère 10.13. Il en aurait été de même pour la propriété CSS `visibility:hidden`; qui rendra le texte aussi invisible aux internautes regardant la page qu'à ceux utilisant une technologie d'assistance.

Retenez donc que les propriétés CSS `display:none`; et `visibility:hidden`; sont à proscrire pour cacher en permanence du texte de façon accessible.

Évidemment, quand il s'agit d'un texte caché, qui n'a pas vocation à être restitué, le critère est non applicable. De même, vous pouvez tout à fait cacher un texte avec `display:none` ou `visibility:hidden` et modifier la valeur de la propriété en JavaScript pour permettre un affichage du texte en fonction d'une action de l'utilisateur, par exemple.

Se repérer dans la page

Quand on navigue sur une page web, nous avons tendance, presque inconsciemment, à chercher du regard tout ce qui est cliquable. Quand on ne voit pas la page, on utilise un lecteur d'écran qui peut fournir la liste des liens. Par contre, quand on n'utilise pas de lecteur d'écran mais qu'on a malgré tout des troubles de la vue, il est impératif de rendre bien visible ce qui est cliquable : soit en soulignant les liens, soit en leur donnant une couleur différente.

Ainsi, le critère 10.6 concerne les « liens dont la nature n'est pas évidente », c'est-à-dire les liens sur lesquels on n'aurait pas idée de cliquer. Un lien présent en pied de page ou dans un menu n'a pas besoin d'être souligné, on sait qu'on va pouvoir cliquer dessus. En revanche, un lien dans un texte qui n'est pas souligné et dont la couleur est proche du texte environnant risque de ne pas être perçu.

AccessiWeb demande à cet égard que « chaque lien texte signalé uniquement par la couleur et dont la nature n'est pas évidente ait un rapport de contraste supérieur ou égal à 3:1 par rapport au texte environnant ».

OUTILS Tester les rapports de couleur avec ColorZilla et Juicy Studio

La formule de calcul impose le même ratio que dans le cadre de la thématique « Couleurs » : un rapport de 3:1. Sauf que cette fois-ci, il faut faire le calcul non pas entre la couleur de premier plan et celle d'arrière-plan, mais entre la couleur du lien et celle du texte qui se trouve autour.

Pour commencer, il vous faut sélectionner la couleur du texte et celle du lien. Pour cela, vous pouvez regarder le code grâce à Firebug ou récupérer le code couleur avec l'extension Firefox ColorZilla.

► <http://ur1.ca/a6b70>^a

Ensuite, rendez-vous sur le site de Juicy Studio Luminosity Colour Contrast Ratio Analyser.

► <http://ur1.ca/a6b7c>^b

The screenshot shows the 'Analyse Luminosity Contrast Ratio' tool. It has a section for 'Colours' with two input fields: 'Background Colour' set to '#fff' and 'Foreground Colour' set to '#000'. Below these is a 'Calculate Luminosity Contrast Ratio' button. Underneath, there are two text samples: 'Large Text Sample' and 'Regular Text Sample'. The 'Large Text Sample' is shown in a large, bold, black font, while the 'Regular Text Sample' is in a smaller, regular black font. At the bottom, the 'Results for Luminosity Contrast Ratio' section displays 'The contrast ratio is: 21.00:1' and a status message: 'Passed at Level AAA: The luminosity contrast ratio is very good for the chosen colours (#fff and #000).'

Figure 10–15 Capture d'écran de l'outil d'analyse du contraste des couleurs et de la luminosité de Juicy Studio

Il vous suffit d'entrer le code hexadécimal de la couleur du lien dans l'un des deux champs, puis celui du texte environnant dans l'autre, et de valider. L'outil analysera le rapport de contraste et vous l'indiquera.

a. <https://addons.mozilla.org/fr/firefox/addon/colorzilla/>

b. <http://juicystudio.com/services/luminositycontrastratio.php#specify>

Voyons un exemple concret avec le site Easyvol.fr (figure 10-16) : sur cet extrait de la page d'accueil, les liens ne sont pas visibles.

Figure 10-16

Capture d'écran d'une section de la page d'accueil du site Easyvol (<http://www.easyvols.fr>)



Maintenant, observons l'encart intitulé « Comparer et trouver votre billet d'avion en 1 seul clic ».

Figure 10-17

Zoom sur l'encart « Comparez et trouvez votre billet d'avion en 1 seul clic »



Vous voyez le paragraphe... Eh bien, il contient un lien. Vous ne me croyez pas ? Regardons ce qui se passe quand on tabule pour naviguer au clavier.

Le fin liseré pointillé autour du mot voyage nous indique qu'il s'agit bien d'un lien. Or, il est difficile de le distinguer du texte environnant, car il a pratiquement la même couleur et ne dispose pas d'un signe distinctif qui donnerait l'indication qu'il est cliquable, comme un soulignement. Il s'agit d'un cas typique d'invalidation du critère 10.6. La

10 – Éléments obligatoires, structuration de l'information et présentation de l'information

Figure 10-18

Aperçu du focus sur le lien
« voyage » dans le texte de
l'encart

2

Comparer et trouver votre billet d'avion en 1 seul clic

Nous comparons la quasi-intégralité du marché français afin de vous offrir un panorama complet des propositions de billets d'avion des acteurs de l'aérien en France. Plusieurs outils sont à votre disposition pour affiner votre recherche : vous pouvez trier les vols par prix, par tranches horaires, choisir un vol avec ou sans escale, ou votre compagnie... Retrouvez toutes les informations pour votre voyage sur Easyvoyage.

nature du lien n'est pas évidente et le rapport de contraste est inférieur à 3:1, sans même avoir besoin de faire un test. Ce lien est invisible pour tout le monde.

Autre règle pour bien se repérer dans la page : permettre à une personne ayant un handicap moteur ne lui permettant pas de naviguer à la souris de repérer son parcours dans la page quand elle navigue au clavier. Il s'agit du critère 10.7. Ce test est très simple à faire, il vous suffit de tabuler en appuyant sur la touche **Tab** pour avancer, et **Shift+Tab** pour revenir en arrière.

Par défaut, on peut suivre le curseur grâce à un fin liseré pointillé qui entoure chaque élément cliquable quand le focus est positionné dessus. C'est ce que nous avons vu dans la capture précédente et ce qui nous a permis d'identifier qu'il y avait un lien.

Voyons un autre exemple avec le site de l'Université de Nice-Sophia-Antipolis (figure 10-19). Sur la capture d'écran, le focus est positionné sur l'onglet **Accueil**, mais on a pourtant l'impression que rien ne se passe. On ne voit pas où l'on se trouve.

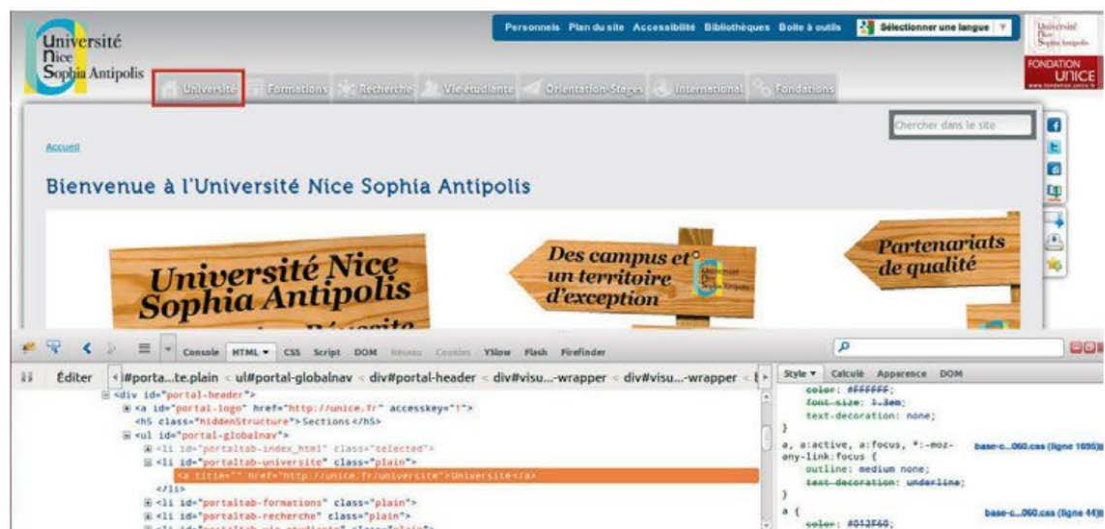


Figure 10-19 Page d'accueil de l'université de Nice-Sophia-Antipolis : quand le focus est sur l'onglet Accueil du menu principal, rien ne se passe. (<http://unice.fr>)

Le paramétrage de ce liseré au focus est fait en CSS grâce à la propriété `outline`. Par défaut, dans les feuilles de styles des navigateurs, `outline` a toujours une valeur, ce qui le rend visible. Si vous ne touchez pas à cette propriété, le critère 10.7 sera validé.

Cependant, il peut arriver d'avoir recours à des feuilles de styles pour remettre à zéro les styles des différents navigateurs, afin de repartir sur une base commune et d'éviter les problèmes de mise en page selon les navigateurs. On parle souvent de fichier `reset.css`. Or, certaines de ces feuilles `reset.css` mettent la propriété `outline` à zéro, ce qui invalide le critère 10.7 et empêche une personne naviguant au clavier de se repérer dans la page.

Il vous faut donc penser, dans ce cas, à surcharger cette feuille de styles en spécifiant une valeur à la propriété `outline`. Vous pouvez même en profiter pour améliorer cette valeur car, même si par défaut le fin liseré pointillé valide le critère 10.7, il faut bien reconnaître que cela reste très discret et peu visible. C'est particulièrement vrai pour les images-liens dont le fin liseré du focus se confond avec la bordure de l'image et rend la navigation clavier particulièrement difficile sur ces éléments.

Sur le site d'ACS Horizons, nous avons amélioré la visibilité de la tabulation (figure 10-20).

Figure 10-20

Focus sur le lien « Accessibilité numérique » : on voit bien mieux où on se situe grâce aux tirets rouges bien visibles. (<http://acs-horizons.fr>)



Voici le code CSS utilisé pour améliorer la visibilité de la prise de focus :

```
a:focus, a:focus img {  
    outline: 0.15em dashed red;  
}
```

C'est simple et efficace : n'hésitez pas à améliorer le focus, cela rendra un grand service à vos utilisateurs naviguant au clavier, qu'ils soient en situation de handicap ou non.

Attention ! Nous vous rappelons que l'information ne doit pas être donnée uniquement par la couleur. Donc si vous décidez de mettre une couleur de fond pour donner de la visibilité à la prise de focus, pourquoi pas ! Mais pensez à proposer également une valeur pour la propriété `outline`.

Lisibilité sur écran et dyslexie

Certaines personnes avec un handicap cognitif, les personnes dyslexiques notamment, suivent plus facilement des yeux un texte quand les blocs de texte ont une taille réduite et quand il y a un espace suffisant entre les lignes et les paragraphes. De même, la lecture est rendue difficile quand l'espacement entre les mots est différent selon les lignes, ce qui est le cas quand on justifie un texte en l'alignant de chaque côté.

Deux recommandations sont faites à ce sujet, toutes deux de niveau Or mais faciles à mettre en place :

- le critère 10.9 demande à ce qu'un texte ne soit pas justifié (c'est-à-dire aligné des deux côtés) ;
- le critère 10.12 demande à ce qu'il y ait un interligne de 1,5 em.

Prenons l'exemple d'une page sur le site des éditions du Seuil (figure 10-21, page suivante). Le texte est justifié, ce qui rend sa lecture difficile.

Puis, voyons ce que donnerait ce même texte aligné à gauche, comme le recommande le critère AccessiWeb 10.9 (voir figure 10-22).

Et enfin, appliquons un interlignage de 1,5 (figure 10-23). Notons que, par défaut, les navigateurs appliquent un interlignage de 1,2. Observons la page modifiée : le résultat respecte désormais ces critères 10.9 et 10.12.

Même s'il s'agit de critères Or, ces mesures sont très faciles à appliquer et amélioreront nettement l'expérience de vos utilisateurs dyslexiques. Alors, si vous avez la possibilité de le faire, ne vous en privez pas !

Figure 10-21

Page décrivant un livre sur le site des éditions du Seuil.
Le texte est justifié des deux côtés.
(<http://ur1.ca/abve6a>)



a. <http://www.seuil.com/livre-9782021054156.htm>

Figure 10-22

Même capture que précédemment, mais le texte est aligné à gauche.



Figure 10-23

Page du Seuil avec le texte aligné à gauche et un interligne de 1,5



Ne pas donner l'information seulement par la forme, la taille ou la position

Nous avons vu dans la thématique « Couleurs » que les critères 3.1 et 3.2 cherchaient à vérifier que l'information n'est pas donnée uniquement par la couleur. Nous trouvons, avec le critère 10.14, le pendant à cette règle, puisqu'il s'agit ici de vérifier que l'information n'est pas donnée uniquement par la forme, la taille ou la position.

Voyons l'exemple du site de la Communauté intercommunale des villes solidaires du Sud de l'Île de la Réunion (figure 10-24, page suivante). Le menu principal est sous forme d'onglets. Le menu actif ou survolé se soulève.

L'information donnée ici par le soulèvement de l'onglet est :

- soit que l'onglet est survolé. Cette information est également donnée par le changement de forme du curseur qui devient une main. Il est clair qu'on survole un lien, le critère est non applicable ;
- soit que l'onglet est actif.

Dans ce dernier cas, l'information permet de se situer dans l'arborescence du site. Cette information n'est-elle donnée que par la position ? Non, puisqu'elle est égale-

Figure 10-24

Page d'accueil du site de la communauté d'agglomérations CIVIS : les menus se soulèvent quand on les survole ou quand ils sont actifs, comme on le voit pour le menu actif « Accueil » et le menu survolé « Éco-tourisme » (<http://www.civis.re>)



ment donnée par la couleur. Mais cette alternative, la couleur, est-elle pertinente ? Non, puisqu'elle ne permet pas aux personnes ne voyant pas la page de se situer dans l'arborescence comme pour ceux qui voient l'onglet actif surélevé.

La solution ? Vous pouvez ajouter un attribut `title` sur le lien actif en indiquant par exemple : `title="Accueil lien actif"`. Vous pouvez également désactiver le lien actif, ce qui validera le critère mais sera peut-être décrié par certains ergonomes. Enfin, l'obligation porte sur la restitution de l'information d'une autre façon.

Une bonne solution serait donc de mettre un fil d'Ariane, ce qui est facile à faire, permet de situer l'utilisateur, profite à tous et sert en plus à valider un critère Or, le 12.8 (thématique 12 « Navigation », voir le chapitre 11). Le problème avec le fil d'Ariane, c'est qu'il ne permet pas de donner l'information à l'endroit même de l'onglet actif, ce qui invalide donc le critère 10.4. Mais il offre une solution alternative de compromis qui, s'il ne suffit pas à valider formellement le critère d'accessibilité, aide l'utilisateur à bien s'orienter, ce qui est le plus important en termes d'accessibilité, finalement.

Formulaires, navigation et consultation (thématiques AccessiWeb 11 à 13)

Les trois dernières thématiques d'AccessiWeb sont consacrées aux formulaires (11), à la navigation (12) et à la consultation de tous les types de contenus sur le Web (13).

Formulaires

« Associer pour chaque formulaire chacun de ses champs à son intitulé, grouper les champs dans des blocs d'informations de même nature, structurer les listes de choix de manière pertinente, donner à chaque bouton un intitulé explicite. Vérifier la présence d'aide à la saisie, s'assurer que le contrôle de saisie est accessible et que l'utilisateur peut contrôler les données à caractère financier, juridique ou personnelles. »

<http://ur1.ca/a0cf8>¹

Un site web peut être à la fois une plate-forme de communication pour donner des informations à destination des internautes, mais c'est également de plus en plus un moyen d'*échanger* avec eux. Et l'un des moyens les plus courants pour cela réside dans l'utilisation de formulaires.

Formulaires de contact bien sûr, mais également formulaires pour créer un compte client, se connecter à ses comptes bancaires sur le site de sa banque, inscrire son enfant à la cantine, demander un acte d'état civil, s'inscrire à l'université, réserver un billet de train... Les exemples ne manquent pas. Les formulaires sont indispensables pour effectuer toutes ces procédures.

Or, lorsqu'un formulaire n'est pas accessible, les risques sont grands d'induire l'internaute en erreur, surtout quand il s'agit d'un internaute aveugle. Et les conséquences

1. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#formulaire

peuvent être très dommageables. Voyons les règles à respecter pour créer un formulaire accessible.

INFO Les critères de la thématique « Formulaires »

La thématique « Formulaires » contient quinze critères, dont neuf de niveau Bronze, trois de niveau Argent et trois de niveau Or.

Nous traiterons de tous les critères présents dans la thématique.

Étiquettes et boutons : des intitulés pertinents

La première chose à vérifier, c'est que chaque élément de formulaire est identifié de façon pertinente. Nous pensons à la fois aux libellés (ou étiquettes) des champs de formulaire (balise `<label>`), mais également aux intitulés des boutons.

RESSOURCE Vidéo pour des formulaires accessibles

Que se passe-t-il quand une personne aveugle navigue dans un formulaire non accessible ? La réponse en vidéo avec ce petit film édité par l'université de Nice-Sophia-Antipolis dans le cadre du projet Access-Key. On y voit une jeune femme aveugle utilisant une synthèse vocale pour naviguer dans un formulaire non accessible. Elle cherche à répondre à un exercice de questions à choix multiples, avec des cases à cocher. Elle commence par lire toutes les réponses proposées, en naviguant de gauche à droite. Puis elle revient en arrière pour vérifier certaines réponses et cocher celles qu'elle estime correctes. Or, en revenant en arrière, comme les intitulés de réponses ne sont pas liés aux cases à cocher, la synthèse vocale induit la jeune femme en erreur.

Une vidéo très instructive à visionner absolument !

► <http://ur1.ca/a6y9m³>

a. <http://unice.fr/access-key/videos/des-formulaires-accessibles>

Commençons par étudier les étiquettes des champs de formulaires.

Des étiquettes présentes, pertinentes et associées aux champs

Chaque champ de formulaire (balises `<input>`, `<textarea>` et `<select>`) doit être associé à une balise `<label>` venant décrire l'objet du champ. On relie la balise `<label>` à la balise `<input>` grâce aux attributs `id` et `for`. Il faut également veiller à ce que l'étiquette soit accolée au champ qu'elle décrit (critère 11.4).

Voici un exemple d'utilisation :

```
<label for="search">Recherche</label>
<input id="search" name="search" type="text" />
```

Il est également possible de donner une étiquette au champ sans utiliser la balise `<label>`. Cette technique peut-être utile notamment si vous ne souhaitez pas qu'un texte apparaisse à l'écran pour identifier le champ. Il suffit alors d'utiliser l'attribut `title` sur la balise `<input>`.

Reprenons notre exemple :

```
<input title="Recherche" name="search" type="text" />
```

Notons que cette technique permet de valider le critère 11.1, mais qu'il reste préférable *d'utiliser une balise `<label>`*, car cela permet à tous de voir l'intitulé du champ. En effet, le `title` est avant tout prévu pour les utilisateurs de lecteurs d'écran, bien qu'une infobulle apparaisse au survol pour les personnes naviguant à la souris (et toujours rien pour les personnes naviguant exclusivement au clavier).

Une fois l'étiquette associée au champ, il reste bien sûr à vérifier que l'intitulé est pertinent.

Prenons un exemple réel. Sur le site de la ville de Saint Parres aux Tertres, plusieurs formulaires sont mis à disposition pour effectuer différentes actions, comme inscrire son enfant à la cantine (figure 11-1).

Figure 11-1
Formulaire d'inscription
à la cantine sur le site
de la ville de Saint Parres
aux Tertres
(http://ur1.ca/a9zdr^a))
(suite pages suivantes)

The image shows a web form titled "Formulaire à remplir" (Form to be filled) with a subtitle "Fiche d'inscription au restaurant scolaire" (School restaurant registration card). The form is set against a light green background and contains several input fields for personal information. It is organized into three main sections: one for the child, one for the father, and one for the mother. Each section includes fields for name, address, and phone number. The form is presented in a clean, structured layout with horizontal lines separating the different sections.

Formulaire à remplir	
Fiche d'inscription au restaurant scolaire	
Nom et prénom de l'enfant	
Date de naissance	
Ecole et Classe fréquentées (Année 2012-13)	
Nom et prénom du père	
Adresse du père	
N° de téléphone	
N° de portable	
Nom et prénom de la mère	
Adresse de la mère	
N° de téléphone	

a <http://www.saintparresauxtertres.fr/inscription-cantine.htm>

N° de portable

Responsable(s) de l'enfant
(en cas de divorce ou de séparation)

Adresse électronique

Profession du père

Nom et adresse de l'employeur

N° de téléphone de l'employeur

Profession de la mère

Nom et adresse de l'employeur

N° de téléphone de l'employeur

N° d'allocataire CAF

Nom, adresse et téléphone d'une
personne à contacter en cas d'urgence

Horaires de fréquentation : (vous devez cocher toutes les cases)

JOURS	OUI	NON
LUNDI	☐	☐
MARDI	☐	☐
JEUDI	☐	☐
VENDREDI	☐	☐

Noms et prénoms des personnes pouvant
récupérer l'enfant et lien de parenté avec
l'enfant

Visualisez le règlement intérieur afin de bien comprendre les modalités d'inscriptions :
[+ PDF](#)

En cas d'accident ou de maladie entre 12h et 14h,
votre enfant devra t-il être transporté :

11 – Formulaires, navigation et consultation (thématiques AccessiWeb 11 à 13)

Au Centre Hospitalier de TROYES ☐

A la clinique (nom, adresse, téléphone)

Traitement médical :

J'autorise la prise de médicaments par mon enfant selon les prescriptions médicales. Un double de l'ordonnance devra être fourni ☐

Je n'autorise pas la prise de médicaments par mon enfant ☐

Photos - vidéos :

J'autorise mon enfant à être photographié ou filmé ☐

Je n'autorise pas mon enfant à être photographié ou filmé ☐

Renseignements complémentaires que vous jugez nécessaire de nous faire connaître :

Je soussigné(e)

Père ou mère ou responsable
légal de l'élève

déclare avoir pris connaissance du règlement intérieur
de l'accueil du restaurant scolaire

Date :

Signature du Père ou de la Mère ou du représentant légal :

Rendre le formulaire dûment signé en Mairie

Vous devez imprimer ce
formulaire avant de l'envoyer

Imprimer

Envoyer

Nous avons des champs de texte court et des zones de texte long, des cases à cocher et deux boutons *Imprimer* et *Valider*. Le formulaire est divisé visuellement en neuf zones, que nous étudierons en détail ci-après.

Voyons plus précisément la zone concernant l'identification de l'enfant (figure 11-2).

Figure 11-2

Zoom sur la partie concernant l'identification de l'enfant



Nom et prénom de l'enfant	
Date de naissance	
Ecole et Classe fréquentées (Année 2012-13)	

Voici le code source correspondant à cette partie :

```
<table bgcolor="#cede66" border="1" bordercolor="#8da219" cellpadding="5"
cellspacing="0" width="580" align="center">
<tbody>
<tr>
<td valign="top" align="center">
<table border="0" cellpadding="5" cellspacing="0" width="570">
<tbody>
<tr class="texte" valign="middle">
<td class="texte-gras" width="285" align="right">
Nom et prénom de l'enfant
</td>
<td width="285" align="left">
<input class="texte-cadre" name="nom_enfant" id="nom_enfant" size="40"
type="text">
</td>
</tr>
</tbody>
</table>
<table border="0" cellpadding="5" cellspacing="0" width="570">
<tbody><tr class="texte" valign="middle">
<td width="285" align="right">
Date de naissance
</td>
<td width="285" align="left">
<input class="texte-cadre" name="date_naissance" id="date_naissance" size="25"
maxlength="75" type="text">
</td>
</tr>
</tbody>
</table>
<table border="0" cellpadding="5" cellspacing="0" width="570">
<tbody>
```

```
<tr class="texte" valign="middle">
<td width="285" align="right">
Ecole et Classe fréquentées (Année 2012-13)
</td>
<td width="285" align="left">
<input class="texte-cadre" name="ecole_classe" id="ecole_classe" size="40"
type="text">
</td>
</tr>
</tbody>
</table>
```

Nous voyons dans cet exemple trois champs de texte court avec un intitulé, mais celui-ci n'est pas identifié comme tel ni relié au champ. Pourtant, les champs de texte (balise `<input>`) ont déjà un attribut.

Voyons comment rendre ce formulaire plus accessible : commençons par supprimer la mise en page en tableau et par enlever tous les attributs de mise en page, que nous placerons dans les CSS afin de rendre le code plus accessible et plus lisible.

```
<label for="nom_enfant">Nom et prénom de l'enfant</label>
<input class="texte-cadre" name="nom_enfant" id="nom_enfant" size="40"
type="text">

<label for="date_naissance">Date de naissance</label>
<input class="texte-cadre" name="date_naissance" id="date_naissance" size="25"
maxlength="75" type="text">

<label for="ecole_classe">École et Classe fréquentées (Année 2012-13)</label>
<input class="texte-cadre" name="ecole_classe" id="ecole_classe" size="40"
type="text">
```

Grâce à ce code à la fois plus simple puisque ne mélangeant plus la forme et le fond, et contenant des balises `<label>` reliées aux champs texte, il n'y a plus de confusion possible en naviguant dans ce formulaire.

Une fois passé en mode formulaire, l'utilisateur de lecteur d'écran n'aura plus accès qu'aux seuls champs de formulaires, les intitulés étant repris seulement s'ils sont associés. Il pourra ainsi tabuler pour entrer dans le champ et la synthèse vocale (ou sa plage Braille) lui lira : « édition nom et prénom de l'enfant », puis en tabulant pour se rendre dans le champ de formulaire suivant, le lecteur d'écran lira : « édition école et classe fréquentées (année 2012-13) ». Le champ est donc associé au `<label>`, il n'y a plus aucune confusion possible.

Il faut veiller par ailleurs à ce qu'un champ ayant la même fonction sur un ensemble de pages (ou sur une même page, mais c'est plus rare) dispose d'un intitulé cohérent (critère 11.3, de niveau Argent). Il est, par exemple, recommandé de ne pas changer l'intitulé du champ de recherche d'une page à l'autre dans un même ensemble de pages.

Des boutons à l'intitulé pertinent

Outre les étiquettes, il faut s'assurer que les boutons ont un intitulé pertinent. Cela peut se faire soit en remplissant l'attribut `value` d'une balise `<input>` de type `submit`, `reset` ou `button`, soit en vérifiant la pertinence du texte contenu dans la balise `<button>`, soit encore en vérifiant le contenu de l'attribut `title` du bouton.

Un intitulé pertinent permet de donner une indication de ce qui va se passer quand on clique sur le bouton.

Voyons l'exemple du formulaire de création d'un compte sur le site [mon.Service-Public.fr](http://mon.service-public.fr) (figure 11-3). À l'issue du renseignement des champs sur le formulaire, vous avez le choix entre deux boutons : *Valider* ou *Annuler*.

Figure 11-3

À l'issue du renseignement des champs sur le formulaire, vous avez le choix entre deux boutons : *valider* ou *annuler*. (<http://ur1.ca/aagmea>)

CRÉATION DE MON COMPTE

Le symbole * est indiqué devant les champs obligatoires.

* Identifiant : De 4 à 30 caractères alphanumériques.

* Mot de passe : De 6 à 20 caractères, avec au moins une lettre et un chiffre

* Répétez votre mot de passe :

* Adresse électronique : Exemple : antoine@free.fr, francois@hotmail.com

☐ Je désire recevoir les informations des organismes partenaires de mon.service-public.fr dans ma boîte électronique.

☐ Je désire recevoir les actualités mon.service-public.fr dans ma boîte électronique.

* Recopier les caractères de l'image :

[Changer l'image](#)
Vous êtes mal-voyant ?

☐ J'accepte les conditions générales d'utilisation

A quoi ça sert ?
Votre identifiant et votre mot de passe vous permettent d'accéder à votre compte. Ces informations sont personnelles, ne les transmettez à personne !

A quoi ça sert ?
Cette étape nous permet d'empêcher les systèmes automatisés de création de compte d'opérer. Merci de bien vouloir saisir dans la zone dédiée, la suite de caractères affichée.

a. <https://creation.mon.service-public.fr/creation/faces/jsp/saisieCreation.jsp>

Le code pour ces deux boutons est le suivant :

```
<span class="inputButtonBefore marginRight10">  
<input id="corps:saisieCreation:btnValiderCreationCpt"  
name="corps:saisieCreation:btnValiderCreationCpt" value="Valider"  
class="inputButton" type="submit">  
</span>
```

```
<span class="inputButtonBefore">
<input id="corps:saisieCreation:btnAnnulerCreationCpt"
name="corps:saisieCreation:btnAnnulerCreationCpt" value="Annuler"
class="inputButton" type="submit">
</span>
```

Les deux boutons sont bien codés comme des boutons, puisqu'il s'agit de balises `<input>` avec l'attribut `type="submit"`. Le critère 11.9 est applicable, les lecteurs d'écran les restitueront donc comme des boutons.

Reste à savoir si l'intitulé des boutons est pertinent. Dans le cas présent, et *uniquement parce que la page ne contient qu'un seul formulaire*, les valeurs `Valider` et `Annuler` dans l'attribut `value` sont claires et permettent de savoir quelle action va produire une activation des boutons. Le critère 11.9 est validé. En revanche, s'il y avait eu un autre formulaire sur la page, il aurait fallu être plus précis, en intitulant par exemple le bouton de validation « Créer le compte ».

ATTENTION Éviter les intitulés génériques

Notez que les intitulés « valider », « OK », « envoyer », etc. sont beaucoup trop vagues et ne permettent pas de savoir réellement quelle action est effectuée en appuyant sur le bouton.

S'agit-il d'une recherche ? Un bouton *Rechercher sur le site* est plus clair. S'agit-il de s'abonner à la lettre d'informations ? Préférez *Je m'abonne* à un bouton *Envoyer*.

En plus d'être nécessaire pour éviter de perdre des internautes en situation de handicap, il est bon de savoir qu'un intitulé trop générique sur des boutons d'action peut faire perdre des ventes sur un site commercial, par exemple, les internautes hésitants ayant besoin d'être rassurés en sachant précisément ce qu'implique un clic sur le bouton. Et ce, en dehors même d'une question de handicap. Pensez-y, ce serait dommage de perdre des visiteurs pour une simple histoire d'intitulé de bouton.

Si nous reprenons à présent notre exemple concernant l'inscription à la cantine pour la ville de Saint Parres aux Tertres, nous voyons qu'il y a deux boutons : l'un pour imprimer le formulaire, l'autre pour le valider (figure 11-4).

Figure 11-4

Les deux boutons pour imprimer et valider le formulaire d'inscription à la cantine sur le site de la ville de Saint Parres aux Tertres

Rendre le formulaire dûment signé en Mairie

Vous devez imprimer ce formulaire avant de l'envoyer

Imprimer

Envoyer

Voyons le code pour ces deux boutons :

```
<td width="125" align="center">
<a href="javascript:onclick=window.print();">
```



```


</a>
</td>
<td width="125" align="center">
<input name="envoyer" id="envoyer" src="img-elements-types/btn-
envoyer.gif" alt="Envoyez le formulaire" height="37" border="0"
type="image" width="94">
</td>

```

Le problème avec ce formulaire, c'est que les deux boutons n'en sont pas. Le premier est simulé via un lien associé à du JavaScript. Le texte du bouton est en fait restitué via l'attribut `alt` sur l'image. Ce sont les thématiques « Images » et « Scripts » qui sont concernées. Quant au second bouton, c'est un champ de formulaire, mais de type image. Ce genre de champ est traité dans la thématique « Images ». C'est l'attribut `alt` qui sera restitué.

Bref, dans le cas de ce formulaire, le critère 11.9 relatif à l'intitulé des boutons est non applicable, puisqu'il n'y a aucun bouton structuré comme tel en HTML.

Voyons le cas maintenant où plusieurs boutons à l'intitulé identique sont présents sur une même page, avec le site de la ville de Lyon (figure 11-5).

Les trois formulaires concernent respectivement l'affichage d'informations pratiques (numéros d'urgence, pharmacies de garde, etc.), le choix d'une rubrique concernant les démarches administratives et la recherche d'un événement dans l'agenda. Notons que le moteur de recherche a également un bouton de validation intitulé « OK ».

À première vue, un intitulé « OK » sur des boutons servant à effectuer plusieurs actions différentes n'est pas pertinent, car il n'identifie pas l'action.

Voyons le code des boutons ayant pour intitulé « OK » :

```

<!--Recherche-->
<input class="submit" value="OK" id="submit" type="submit">

<!--Lyon pratique-->
<input value="Ok" id="submit_droite1p" class="submit_droite"
onclick="clientRedirect(document.getElementById('select_lyon_pratique'))
);" type="button">

<!--Vos démarches-->
<input value="Ok" id="submit_droitevdm" class="submit_droite"
onclick="clientRedirect(document.getElementById('select_vos_demarches'))
);" type="button">

```



```
<!--Agenda-->
<input id="Rechercher" name="Rechercher" value="Ok"
class="submit_droite" type="submit">
```

Nous constatons qu'il s'agit bien de boutons de formulaires (balises `<input>` avec `type="submit"` ou `type="button"`). Et la valeur de chaque bouton est bien identique. Le critère 11.9 est invalidé.

Figure 11-5

Sur la page d'accueil de la ville de Lyon, on remarque trois formulaires sur la barre latérale droite, avec un bouton intitulé « OK ». (<http://www.lyon.fr>)

The screenshot displays the right sidebar of the Lyon.fr website, featuring three distinct form sections:

- Lyon pratique:** Includes icons for a plus sign, a telephone, and a windmill. Below them is a search bar labeled 'Numéros d'urgence' with an 'OK' button.
- Vos démarches:** Features a 'LYONendirect' logo and a list of services: 'Acte de naissance', 'Vignette pour stationnement résidentiel', 'Calcul du quotient familial', 'Passeport biométrique', 'Organiser un événement sur l'espace public', and 'Offres d'emplois Ville de Lyon'. At the bottom is a search bar labeled 'Emploi' with an 'OK' button.
- Agenda:** Includes a search bar with an 'OK' button and a '+' icon. Below it are links for 'Aujourd'hui', 'Cette semaine', and 'Ce mois'.

Comment faire pour rendre ce critère valide sans dénaturer le graphisme qui nécessite des intitulés courts et identiques avec un bouton **OK** ? La solution consisterait à ajouter des attributs `title` pertinents, comme ci-dessous :

```
<!--Recherche-->
<input class="submit" value="OK" title="Rechercher dans le site"
id="submit" type="submit">

<!--Lyon pratique-->
<input value="Ok" title="Afficher les infos pratiques"
id="submit_droite1p" class="submit_droite"
onclick="clientRedirect(document.getElementById('select_lyon_pratique'));" type="button">

<!--Vos démarches-->
<input value="Ok" title="Choisir une rubrique pour les démarches"
id="submit_droitevdm" class="submit_droite"
onclick="clientRedirect(document.getElementById('select_vos_demarches'));" type="button">

<!--Agenda-->
<input id="Rechercher" name="Rechercher" value="Ok" title="Rechercher
un événement dans l'agenda" class="submit_droite" type="submit">
```

Structurer ses formulaires

Vous avez des étiquettes et elles sont bien liées aux champs ? Parfait ! Maintenant, passons à la structuration des informations dans un formulaire. Il s'agit de regrouper les informations de même nature et de structurer les listes de choix.

Regrouper les informations de même nature

Les critères 11.5 et 11.6 visent à regrouper les informations de même nature grâce à l'utilisation des balises `<fieldset>` (qui regroupent) et `<legend>` (qui donnent un intitulé au regroupement). Notons que la balise `<legend>` est systématiquement restituée par les lecteurs d'écran quand elle est présente. Il s'agit bien de regrouper les informations de même nature, et non d'imposer des regroupements, même s'ils vous paraissent logiques.

Chaque personne a un schéma mental personnel d'organisation des informations. S'intéresser à l'accessibilité ne signifie pas imposer un schéma mental, mais simplement rendre accessible à tous les informations telles qu'elles ont été pensées par l'auteur.

Ainsi, des champs de date (jour, mois, année) sont de même nature et doivent impérativement être regroupés. De même, dans un questionnaire, si plusieurs boutons

radio « oui/non » sont présents, ils doivent être regroupés dans un `<fieldset>` dont la légende sera alors la question posée.

Reprenons notre exemple de formulaire d'inscription à la cantine (figure 11-1, pages 231 à 233). On y voit simulés visuellement plusieurs blocs d'informations.

Le premier bloc, encadré par une bordure et avec un fond vert, entoure les informations relatives à l'identité de l'enfant et les informations de contact des parents. Des sections sont simulées là encore visuellement à l'intérieur avec un sous-bloc concernant l'enfant, un autre concernant le père et le dernier concernant la mère.

Un second bloc, sur fond gris, regroupe les informations relatives à la profession des parents.

Puis, des sections sont simulées en-dessous par des titres. Simulées seulement, puisque le code (`<span class="titre-bleu">Horaires de fréquentation : </span>`) n'utilise pas de balise de titre `<h>`.

De nombreux critères seront invalidés pour cette page, mais peut-on considérer que ces blocs visuellement identifiés doivent l'être également par l'utilisation de balises `<fieldset>` et `<legend>` ?

C'est incontestablement une bonne pratique, mais la recommandation d'AccessiWeb va dans le sens d'une interprétation souple des critères 11.5 et 11.6. La seule question à se poser, en dehors de l'organisation visuelle, est la suivante : y a-t-il des informations de même nature qui doivent impérativement être regroupées pour faire sens ?

Ainsi, dans notre exemple, il ne sera pas forcément nécessaire d'identifier les blocs visuels avec `<fieldset>`. En revanche, certains champs ont le même intitulé. Par exemple, les champs *N° de téléphone*, *N° de portable*, *Nom et adresse de l'employeur* sont identiques pour le père et la mère. Il est obligatoire de regrouper ces champs dans une balise `<fieldset>` qui aura pour légende respectivement « Père » et « Mère ».

Par contre, les champs *Adresse du père* et *Adresse de la mère* sont suffisamment explicites et n'auraient pas besoin d'être encadrés par un `<fieldset>` pour satisfaire aux critères AccessiWeb.

Voici ce que cela donnerait, par exemple, pour le premier bloc visuel sur fond vert concernant le père :

```
<label for="nom_pere">Nom et prénom du père</label>
<input type="text" id="nom_pere" name="nom_pere" />
<label for="adresse_pere">Adresse du père</label>
<textarea id="adresse_pere" name="adresse_pere"></textarea>
<fieldset>
<legend>Coordonnées du père</legend>
<label for="tel_pere">N° de téléphone</label>
```

```
<input type="text" id="tel_pere" name="tel_pere" />
<label for="mob_pere">N° de portable</label>
<input type="text" id="mob_pere" name="mob_pere" />
</fieldset>
```

Ce code suffit à valider les critères AccessiWeb 11.5 et 11.6, mais nous pouvons évidemment faire beaucoup mieux !

Voici comment restructurer l'exemple précédent de façon plus optimale :

```
<fieldset>
<legend>Coordonnées du père</legend>
<!--on supprime la précision "père" sur les champs nom et adresse,
puisqu'il est évident que la légende du fieldset est suffisamment explicite-->
<label for="nom_pere">Nom et prénom</label>
<input type="text" id="nom_pere" name="nom_pere" />
<label for="adresse_pere">Adresse</label>
<textarea id="adresse_pere" name="adresse_pere"></textarea>
<label for="tel_pere">N° de téléphone</label>
<input type="text" id="tel_pere" name="tel_pere" />
<label for="mob_pere">N° de portable</label>
<input type="text" id="mob_pere" name="mob_pere" />
</fieldset>
```

Il est évident que cette solution est meilleure, mais elle n'est pas plus accessible. Attention donc à ne pas vous tromper si vous faites une évaluation en accessibilité. Rien ne vous empêche de proposer des solutions plus efficaces, comme avec notre seconde proposition ci-dessus, mais vous ne devez pas pour autant invalider les critères 11.5 et 11.6 si la solution proposée ressemble à notre première proposition.

Structurer les listes de choix

Sur le même principe de structuration, il vous faudra penser à structurer de façon pertinente vos listes de choix introduites par la balise `<select>` - là encore, seulement quand c'est nécessaire.

Prenons l'exemple de la liste déroulante sur le site Fromages.com (figure 11-6). Quand on ouvre la liste pour choisir le pays de livraison, on voit une structuration visuelle avec cinq pays regroupés, puis dessous, la liste des autres pays par ordre alphabétique.

Or, cette liste de choix n'est pas structurée de manière pertinente pour l'utilisateur. Elle l'est sans doute pour le vendeur, qui a probablement voulu mettre en tête de liste les pays vers lesquels ils vendent le plus, mais rien n'indique une quelconque logique d'un point de vue de la structure dans le code HTML.

Figure 11-6

Choix du pays de livraison sur le site Fromages.com
(<http://ur1.ca/aa8cl>)



a. <http://www.fromages.com/fr/abonnements>

```
<select id="pays_id" name="pays_id"
onchange="$('#formulaireRechercheAbonnement').submit();">

<!-- l'option ci-dessous n'est pas un choix mais une étiquette pour le
champ select-->
<option value="0">Filtrer par pays</option>

<!-- on voit ici une option vide pour simuler une séparation visuelle-->
<option value=" "></option>

<option value="109">USA</option>
<option value="48">JAPON</option>
<option value="107">GRANDE BRETAGNE</option>
<option value="16">CANADA</option>
<option value="32" selected="selected">FRANCE</option>

<!-- on voit ici une option vide pour simuler une séparation visuelle
entre les deux groupes de pays-->
<option value=" "></option>

<option value="92">AFRIQUE DU SUD</option>
<option value="35">ALLEMAGNE</option>
```



```
<option value="117">Arabie Saoudite</option>
<option value="1">ARGENTINE</option>
<!-- on coupe la liste qui est très très longue-->
</select>
```

Pour corriger cette liste, nous aurions pu soit supprimer ce regroupement visuel et mettre une liste alphabétique simple, soit regrouper effectivement les options avec une balise `<optgroup>` et un attribut `label` pour identifier les regroupements. Ces regroupements peuvent suivre la logique du vendeur ou reprendre les grandes zones géographiques (Europe, Amérique du Nord, Asie, Afrique...). Nous allons ici nous contenter simplement de donner une structure sémantique au regroupement visuel prévu initialement :

```
<label for="pays">Filtrer par pays</label>
<select id="pays_id" name="pays_id"
onchange="$('#formulaireRechercheAbonnement').submit();">

  <optgroup label="Pays fréquemment livrés">
    <option value="109">USA</option>
    <option value="48">JAPON</option>
    <option value="107">GRANDE BRETAGNE</option>
    <option value="16">CANADA</option>
    <option value="32" selected="selected">FRANCE</option>
  </optgroup>

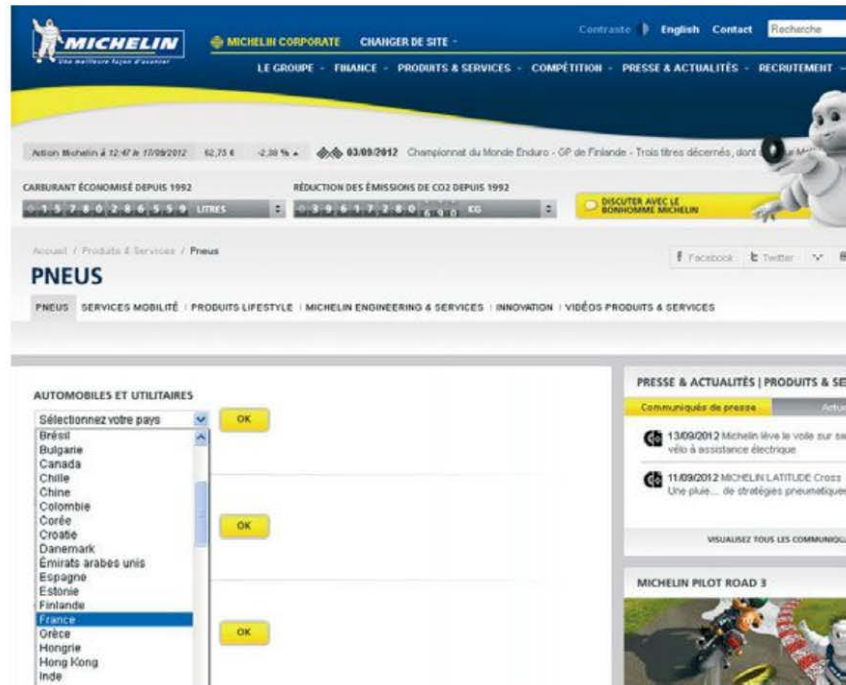
  <optgroup label="Autres pays, par ordre alphabétique">
    <option value="92">AFRIQUE DU SUD</option>
    <option value="35">ALLEMAGNE</option>
    <option value="117">Arabie Saoudite</option>
    <option value="1">ARGENTINE</option>
    <!-- on coupe la liste qui est très très longue-->
  </optgroup>
</select>
```

Remarquez la suppression du choix *Filtrer par pays*, qui ne correspond pas à un choix mais à une étiquette, et son remplacement par une étiquette (la balise `<label>` associée à son champ `select`).

Dans l'exemple du site Michelin (figure 11-7), par contre, le critère 11.8 serait non applicable. En effet, on voit ici que la liste des pays est donnée par ordre alphabétique, sans regroupement. Un regroupement par zones géographiques serait certes possible, mais pas obligatoire. Il n'y a pas besoin d'utiliser de balises `<optgroup>` ni de vérifier la présence d'un attribut `label` pertinent.

Figure 11-7

Choix du pays dans la liste déroulante du site Michelin
(<http://ur1.ca/aa8h9a>)

The image is a screenshot of the Michelin website's corporate page. At the top, there's a navigation bar with the Michelin logo, 'MICHELIN CORPORATE', and links like 'CHANGER DE SITE', 'Contraste', 'English', 'Contact', and 'Recherche'. Below this, a secondary navigation bar lists 'LE GROUPE', 'FINANCE', 'PRODUITS & SERVICES', 'COMPÉTITION', 'PRESSE & ACTUALITÉS', and 'RECRUTEMENT'. The main content area features a section titled 'PNEUS' (Tires) with a sub-header 'PNEUS SERVICES MOBILITÉ | PRODUITS LIFESTYLE | MICHELIN ENGINEERING & SERVICES | INNOVATION | VIDÉOS PRODUITS & SERVICES'. On the left, there's a dropdown menu labeled 'Sélectionnez votre pays' (Select your country) with a list of countries including Brésil, Bulgarie, Canada, Chili, Chine, Colombie, Corée, Croatie, Danemark, Émirats arabes unis, Espagne, Estonie, Finlande, France (highlighted), Grèce, Hongrie, Hong Kong, and Inde. To the right of the dropdown are three 'OK' buttons. On the right side of the page, there's a 'PRESSE & ACTUALITÉS | PRODUITS & SERVICES' section with a 'Communiqués de presse' (Press releases) subsection, showing two recent press releases from 2012. At the bottom right, there's a 'MICHELIN PILOT ROAD 3' section with an image of a motorcycle race.

a. <http://www.michelin.com/corporate/FR/produits/pneus>

Encadrer la saisie des données

Votre formulaire est accessible. L'utilisateur aura peut-être à entrer des données. Il reste donc à vérifier si cela peut se faire de façon accessible : ce sont les critères 11.10 à 11.15 qui le vérifient.

Accompagner la saisie des données

Quand un internaute doit entrer des données dans un formulaire, un cadre précis est parfois attendu pour le traitement. Certains champs sont obligatoires, les dates doivent parfois être données dans un format particulier, par exemple avec l'année sur quatre chiffres (JJ/MM/AAA) et non sur deux chiffres (JJ/MM/AA), etc.

Accompagner l'utilisateur dans sa saisie vise à minimiser les risques d'erreurs, qui sont autant de risques de perturber votre internaute en situation de handicap. Et cela permet également un gain de temps dans le traitement des données.

Le contrôle de saisie

Le contrôle de saisie est l'ensemble des processus qui permettent de prévenir l'utilisateur des erreurs de saisie dans un formulaire.

Le glossaire AccessiWeb précise que le contrôle de saisie d'un formulaire est dit accessible quand « les champs obligatoires et les erreurs de saisie sont signalés de manière accessible (par exemple : les champs obligatoires sont signalés par un symbole textuel et pas seulement par la couleur comme “les champs en rouge sont obligatoires”) ».

Le critère 11.10, de niveau Bronze, demande à ce que chaque formulaire utilise les contrôles de saisie de manière pertinente. Il s'agit de vérifier que les champs obligatoires sont bien mentionnés, pas seulement par la couleur, mais également par une indication qui sera perceptible par ceux ne voyant pas les couleurs (une étoile, par exemple).

Figure 11-8
Formulaire de préinscription
à la piscine sur le site de Stella
Sport Saint Maur
(<http://ur1.ca/aacw6a>)

a. <http://www.stella-natation.com/formulaire-de-pre-inscription.html>

Les erreurs de saisie doivent être indiquées de manière pertinente et le type de données attendu ou le format doivent également être renseignés de façon accessible.

On voit sur la figure 11-8 un formulaire de préinscription à la piscine sur le site de Stella Sport Saint Maur. Le formulaire comporte des champs obligatoires mentionnés par une étoile. Il s'agit bien d'un contrôle de saisie, puisque si les champs ne sont pas renseignés, le formulaire ne pourra pas être traité.

Or, pour valider le critère 11.10, ce contrôle doit être pertinent. Cela pourrait être le cas si la mention indiquant « (*) Les champs marqués d'un astérisque sont obligatoires » figurait avant le formulaire, et non après. Un utilisateur de lecteur d'écran ne lira cette indication qu'après avoir rempli les champs, s'il complète le formulaire au fur et à mesure, ne pouvant avoir une vision d'ensemble de la page. Ce contrôle n'est donc pas pertinent puisqu'il n'est pas pleinement accessible. La correction est toutefois très simple, il suffit de mettre la mention au début du formulaire.

Par ailleurs, un contrôle de saisie pourrait être prévu pour indiquer le format de la date attendu pour le champ *Date de naissance*. Mais aucun contrôle n'est effectué sur ces champs et une erreur n'empêchera pas le traitement du formulaire.

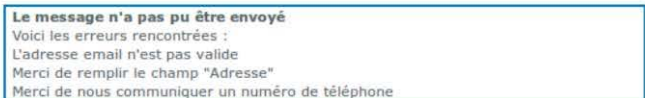
Cela n'invalidera donc pas le critère 11.10, qui cherche à vérifier uniquement les cas où le formulaire utilise effectivement un contrôle de saisie. S'il n'en utilise pas, le critère est non applicable. Mais s'il en utilise, le contrôle doit être utilisé de manière pertinente, c'est-à-dire accessible.

Le critère 11.11, de niveau Argent, demande, quant à lui, à ce que le contrôle de saisie soit accompagné, si possible, de suggestions facilitant la correction des erreurs.

Voyons ce qui se passe si nous ne remplissons pas certains des champs obligatoires sur le site de Stella Sports Saint Maur (figure 11-9). Un message d'erreur placé au début du formulaire indique : « Le message n'a pas pu être envoyé. Voici les erreurs rencontrées : L'adresse email n'est pas valide ; Merci de remplir le champ "Adresse" ; Merci de nous communiquer un numéro de téléphone. »

Figure 11-9

Message d'erreur placé au début du formulaire de préinscription à la piscine sur le site de Stella Sport Saint Maur



Le message n'a pas pu être envoyé
Voici les erreurs rencontrées :
L'adresse email n'est pas valide
Merci de remplir le champ "Adresse"
Merci de nous communiquer un numéro de téléphone

Vos coordonnées

Cette indication en début de formulaire est parfaitement explicite et accessible à tous. Le critère 11.11 sera validé.

Notons qu'il aurait été nécessaire de préciser que des erreurs ont empêché l'envoi du formulaire dès le titre de la page (balise `<title>` située entre les balises `<head></head>`

dans le code HTML). Bien qu'il n'y ait pas de test pour contrôler cela, AccessiWeb fait une obligation d'ajouter dans la page de retour d'erreur la mention « Erreur sur le formulaire » dans le titre de la page.

Cela est spécifique à AccessiWeb, qui va au-delà des recommandations des WCAG, car ce problème est un de ceux qui sont le plus souvent signalés par les utilisateurs. Le problème, si rien n'est mentionné au rechargement de la page, c'est que l'utilisateur de lecteur d'écran peut ne pas penser à vérifier dans le contenu de la page si son formulaire a bien été envoyé.

Mais dans la mesure où aucun test ne le vérifie dans la thématique « Formulaires », cette vérification sera traitée par le critère 8.6 sur la pertinence du titre de la page.

Poursuivons : votre formulaire a des contrôles de saisie accessibles ? Vous pouvez encore améliorer son accessibilité en proposant en plus des aides à la saisie.

Les aides à la saisie

Les aides à la saisies sont encadrées par les critères 11.14 (présence) et 11.15 (pertinence), tous deux de niveau Or. Il s'agit d'apporter à vos utilisateurs un guide pour remplir de façon efficace le formulaire.

Il faut que l'un des moyens suivants soit proposé pour valider le critère 11.14 :

- un lien vers une page d'aide ;
- des indications avant le formulaire ;
- une indication juste avant les champs de formulaires concernés ;
- la présence d'un assistant pour renseigner le formulaire.

Par ailleurs, vous pourrez également proposer, lorsque c'est nécessaire, un correcteur orthographique ou des suggestions de saisie.

Prenons l'exemple d'un site de démonstration proposant la création de formulaires pour la dématérialisation des démarches pour les collectivités territoriales : CapDémat, plate-forme de CapWebCT² (figure 11-10).

L'accessibilité du formulaire a été soignée et des notices pour aider les utilisateurs ont été prévues. Le critère 11.14 est validé. Malheureusement, au moment où nous écrivons ces lignes, il manque encore une ancre au début du formulaire pour pointer vers l'aide. Une aide en bas de formulaire est aussi utile aux personnes ne pouvant pas voir la page dans son ensemble (personnes aveugles et malvoyantes) qu'une mention concernant les champs obligatoires en fin de page. Le critère 11.15 n'est donc pas validé, l'aide à la saisie n'étant pas pertinente.

2. <http://www.capwebct.fr/heading/heading377.html>

Figure 11–10

On voit qu'une notice d'aide détaillée est présente à la fin du formulaire pour en faciliter la compréhension.

Inscription sur les listes électorales

n°63 651

Durée estimée : 5 minutes
Documents à fournir : Aucun

[Abandonner ce brouillon](#) [Continuer plus tard](#)

Inscription

Renseignements sur la personne concernée par la demande

Tous les champs suffixés d'une * sont obligatoires

Sujet *
LEGENDRE Jean

Nationalité *
Française

Raison de la demande *

☒ Domicilié dans la commune ou y habitant depuis six mois au moins

☐ Participe aux contributions directes communales pour la cinquième fois sans interruption

☐ Fonctionnaire public assujéti à résidence obligatoire dans la commune

☐ Ne remplit pas actuellement les conditions d'âge et de résidence mais les remplira d'ici la clôture définitive des listes

Ancienne commune * (Code postal ex:95000)
95000

Confirmer et continuer >

Inscription *

L'étape est complétée 1

Validation *
Avant de pouvoir envoyer votre demande, vous devez compléter toutes les étapes obligatoires.

Aide

Quelles sont les conditions d'accès à ce service ?

La demande d'inscription sur les listes électorales est soumise à plusieurs conditions :

- être majeur,
- être de nationalité française, ou être ressortissant européen,
- habiter la commune ou y payer des impôts,
- jour de ses droits civils et politiques.

Les personnes qui fêtent leur 15^e anniversaire sont inscrites d'office sur les listes électorales de leur commune. Elles reçoivent par courrier une notification de leur inscription. En cas d'absence de notification, elles peuvent effectuer une demande en ligne. La demande d'inscription sur les listes électorales implique que la personne concernée sera radiée des listes de la commune où elle était inscrite auparavant.

Quand s'inscrire sur les listes électorales ?

Les demandes d'inscription sur les listes électorales peuvent être effectuées entre le 1^{er} janvier et le 31 décembre de l'année en cours, l'inscription étant prise en compte au 1^{er} mars de l'année suivante.

Personne à inscrire

Sélectionnez dans la liste l'adulte à inscrire sur les listes électorales. Les adultes déjà inscrits n'y apparaissent pas.

Nationalité

Seules peuvent voter les personnes de nationalité française. Les ressortissants de l'Union européenne résidant en France peuvent toutefois s'inscrire sur les listes complémentaires et voter aux élections municipales et aux élections européennes.

Raison de la demande

- Nouvel électeur La personne concernée s'inscrit pour la première fois sur les listes électorales.
- Domicilié La personne concernée vient d'emménager dans la commune et était auparavant inscrite sur les listes électorales d'une autre commune.
- Contribuable direct La personne concernée n'habite pas la commune mais y possède un bien immobilier pour lequel il paie des impôts locaux depuis au moins 5 ans. Précisez l'adresse du bien dans la commune.
- Résidence obligatoire La personne concernée est un fonctionnaire assujéti à résidence obligatoire dans la commune.

Contrôler le traitement des données

Voyons enfin les critères concernant le contrôle des données par l'utilisateur. Il s'agit des critères 11.12 (niveau Argent) et 11.13 (niveau Or), pratiquement identiques si ce n'est que le champ d'application du critère 11.12 est limité aux données à caractère financier, juridique ou personnel, tandis que le critère 11.13 concerne toutes les données.

Dans les deux cas, il s'agit de permettre à l'utilisateur de modifier, de mettre à jour ou de récupérer ses données. La saisie des données doit permettre, au choix, à l'utilisateur :

- de modifier ou d'annuler les données et les actions effectuées sur ces données après leur saisie ;
- de vérifier et de corriger les données avant la validation du formulaire ;
- de recourir à un mécanisme de confirmation explicite, via un champ de formulaire ou une étape supplémentaire.

Par ailleurs, concernant la suppression des données, l'utilisateur doit pouvoir recourir à un mécanisme pour, au choix :

- récupérer les données supprimées ;
- confirmer la suppression via un champ de formulaire ou une étape supplémentaire.

Concrètement, cela consiste notamment à vérifier que les champs de saisie ne sont pas désactivés automatiquement après la saisie d'une valeur conforme au type et au format attendus. L'utilisateur doit pouvoir contrôler et corriger les données avant la validation du formulaire.

Si ce n'est pas le cas, un mécanisme de confirmation explicite, tel qu'une case à cocher ou une étape de récapitulation, doit permettre à l'utilisateur de confirmer sa saisie. Vous remarquerez que c'est le cas de la plupart des formulaires auxquels on n'a pas appliqué de traitement particulier. Par ailleurs, ces consignes étant de l'ordre de l'amélioration, elles sont bénéfiques pour tous vos utilisateurs, qu'ils soient ou non en situation de handicap. Nous vous encourageons donc à laisser, dans tous les cas, le contrôle de leurs données à vos utilisateurs avant validation.

À NOTER Les formulaires en HTML5

HTML5 crée de nombreux éléments et attributs pour faciliter la vie du développeur, et donc indirectement des visiteurs. Il permet en effet d'harmoniser et de généraliser les contrôles de saisie ou de format, comme le champ date ou l'attribut `required`, par exemple.

C'est d'ailleurs en raison de la « pauvreté » des formulaires que HTML5 a été inventé via le module Web-Form, qui a servi de socle à la création et à l'élaboration de HTML5. Dérivé de XForms, le module Web-Form est le véritable acte de fondation du groupe de travail WHATWG (*Web Hypertext Application Technology Working Group*).

► <http://ur1.ca/abhr0>^a

a. <http://www.whatwg.org/specs/web-forms/current-work/>

Navigation

« Faciliter la navigation dans un ensemble de pages par au moins deux systèmes de navigation différents (menu de navigation, plan du site ou moteur de recherche), un fil d'Ariane et l'indication de la page active dans le menu de navigation. Identifier les groupes de liens importants et la zone de contenu et donner la possibilité de les éviter par des liens de navigation interne. S'assurer que l'ordre de tabulation est cohérent et que la page ne comporte pas de piège au clavier. »

<http://ur1.ca/a0cfc>³

À moins que votre site ne contienne qu'une à trois courtes pages web, vous aurez forcément besoin de naviguer à l'intérieur soit du site, soit de la page via des ancres. Cette thématique est donc particulière, car de nombreux critères portent non sur une page web isolée, mais sur un ensemble de pages ou sur une collection de pages.

Un ensemble de pages est caractérisé par son unicité du point de vue de l'expérience utilisateur. Pour que des pages web soient considérées comme faisant partie d'un ensemble, elles doivent bien sûr être reliées entre elles, mais aussi représenter un ensemble cohérent, tant du point de vue des fonctionnalités ou du contenu que de l'aspect graphique. Par exemple, les pages d'un processus de paiement électronique, les pages d'une rubrique spécifique, les pages d'un blog, les pages d'administration d'un compte client sont autant d'ensembles de pages. AccessiWeb précise que la page d'accueil d'un site web peut constituer, à elle seule, un « ensemble de pages » du fait de son unicité.

Outre la navigation sur un ensemble de pages, il est nécessaire de s'intéresser à la navigation dans une collection de pages. Il s'agit de pages portant sur un même sujet ou ayant une nature commune. C'est typiquement le cas des résultats d'un moteur de recherche ou d'un catalogue.

Une navigation qui ne serait pas claire peut induire en erreur l'utilisateur, lequel peut vite se sentir perdu. Par ailleurs, une navigation accessible doit permettre aux utilisateurs de naviguer rapidement dans les zones importantes, de se repérer dans une navigation cohérente au clavier comme à la souris et de ne pas rester coincés dans la page en raison d'un script non accessible, par exemple.

La thématique « Navigation » est donc tout aussi utile aux personnes en situation de handicap cognitif ou moteur (navigation au clavier) qu'aux personnes malvoyantes ou aveugles.

INFO Critères de la thématique « Navigation »

La thématique « Navigation » contient quatorze critères, dont quatre de niveau Bronze, sept de niveau Argent et trois de niveau Or.

Nous traiterons ici de tous les critères présents dans la thématique.

3. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#navigation

Les systèmes de navigation

Les critères 12.1 à 12.9 portent sur les systèmes de navigation. Un système de navigation est un système permettant de naviguer d'une page ou d'une portion de page à une autre, ou de retrouver une information dans le site. AccessiWeb retient quatre systèmes de navigation :

- le menu de navigation principal ;
- le sommaire interne (ancres menant à différentes sections de la page) ;
- le plan du site ;
- le moteur de recherche.

Dans un ensemble de pages : navigation principale, plan du site, recherche

Le principe est simple, à savoir que tout ensemble de pages doit posséder au moins deux des systèmes de navigation suivants : un menu principal, un plan du site et/ou un moteur de recherche.

La présentation du menu doit être cohérente, c'est-à-dire que celui-ci doit permettre de naviguer dans les différentes rubriques du site de façon logique et doit toujours rester à la même place. Il en va de même pour le moteur de recherche ou le plan du site, qui doivent être accessibles de la même façon dans l'ensemble de pages.

Le plan du site, s'il est présent, doit bien sûr refléter l'organisation du site.

Dans une collection de pages : liens suivant/précédent

Quand il s'agit de naviguer dans une collection de pages, il faut prévoir des liens pour se rendre aux pages suivantes et précédentes, ainsi que des systèmes (liens ou liste déroulante) pour atteindre chacune des pages de résultats. Quand le nombre de résultats est trop important, un regroupement permettant de naviguer toutes les dix pages, par exemple, est autorisé.

Figure 11-11
Exemple de navigation
dans la collection de pages
des résultats de Google



Optimiser sa navigation : le fil d'Ariane

Les critères 12.8 et 12.9 sont de niveau Or, mais tellement faciles et utiles à tous qu'il serait dommage de s'en priver. Il s'agit de mettre en place un fil d'Ariane, ou *breadcrumbs* en anglais, par allusion aux miettes de pain laissées par le Petit Poucet afin de retrouver son chemin. En français, cela renvoie, dans la mythologie grecque, au fil de

laine laissé par Ariane pour que Thésée puisse ressortir du labyrinthe de Dédale après avoir tué le Minotaure.

L'idée est ici de permettre à l'internaute de remonter le fil de l'arborescence du site, grâce à des liens reprenant le cheminement des pages selon le « rubriquage » prévu dans le site.

Dans l'exemple qui suit (site de l'Inalco, figure 11-12), sous le menu principal, la partie éclairée indique que nous sommes sur la page « Chinois » de la rubrique « Département Chine », elle-même atteinte depuis la page « Choisir sa langue » que l'on trouve sur la page « Étudier à l'Inalco » accessible depuis la page « Accueil ».

Figure 11-12

Exemple de fil d'Ariane sur le site de l'Inalco
(<http://www.inalco.fr>)



La plupart des CMS prévoient une fonction très simple pour créer un fil d'Ariane automatique, alors n'hésitez pas !

Les zones de navigation : liens d'accès rapide et liens d'évitement

Quand on navigue au clavier, il est important de pouvoir accéder à certaines zones de la page rapidement, en évitant le très long menu principal, par exemple, pour aller directement au contenu. C'est important pour gagner du temps, mais aussi pour se situer dans la page.

Prévoir une navigation dans les grandes zones de la page est donc profitable à tous ceux qui ont des difficultés à atteindre le contenu principal rapidement et facilement. Nous pensons aux utilisateurs de lecteurs d'écran qui visitent plusieurs pages sur un même site et souhaitent éviter d'avoir à entendre tous les titres ou liens de navigation identiques sur chaque page avant d'atteindre la zone de contenu principal. Les liens d'évitement leur sont donc utiles, même s'il est vrai que les utilisateurs de lecteurs d'écran ont d'autres options pour naviguer dans la page (de titre en titre, notamment).

Mais cela profite avant tout aux personnes ne pouvant naviguer qu'au clavier, qui pourront ainsi atteindre le contenu en tabulant moins. Sans liens d'accès rapide, il leur faudrait peut-être des douzaines, voire parfois des centaines de manipulations au clavier avant d'atteindre un lien dans la zone principale. Cela peut prendre beaucoup de temps et provoquer de graves douleurs physiques chez certaines personnes. Douleurs qui peuvent nécessiter de prendre des antalgiques puissants, de type morphine, et entraîner des effets secondaires tels qu'une baisse de la concentration, voire une impossibilité totale de naviguer sur le Web.

ACTION Ajouter des rôles ARIA landmark

Nous étudierons plus en détail dans le chapitre 12 (page 284) l'utilisation des rôles ARIA permettant de définir de grandes zones de contenus sur lesquelles il est possible de naviguer.

Ajouter des rôles **landmark** est vraiment facile, prend très peu de temps et peut, à terme, rendre de grands services aux personnes ne pouvant naviguer qu'au clavier. On peut en effet espérer qu'une utilisation massive de ces rôles par les développeurs web incitera les concepteurs de navigateurs à développer des fonctionnalités pour faciliter la navigation au clavier, comme le proposent les lecteurs d'écran. Ceci améliorerait le quotidien des personnes ne pouvant pas naviguer à la souris. Alors, à vous de jouer !

Les liens d'accès rapide bénéficient également aux personnes utilisant un logiciel d'agrandissement, qui n'auront ainsi pas à chercher dans des blocs d'informations ou des titres identiques où commence le contenu à chaque fois qu'ils entreront sur une nouvelle page.

Enfin, les personnes présentant des limitations cognitives bénéficieront aussi d'un regroupement des liens dans des zones identifiées.

En résumé : des petits liens pour les développeurs, un grand pas pour les utilisateurs !

D'un point de vue technique, cela se réalise on ne peut plus simplement, puisqu'il suffit d'insérer des points d'ancrage dans la page (critère 12.12) et de créer des ancres pointant vers ces zones (critère 12.13).

Ces liens sont appelés « liens d'évitement » ou « liens d'accès rapide » puisqu'ils permettent, selon les points de vue, de sauter les parties de la page déjà connues, c'est-à-dire d'aller directement à la zone de contenus.

Rien ne vous interdit de cacher ces liens, à condition qu'ils soient accessibles et visibles au focus. C'est ce qui est mis en place, par exemple, sur le site Légifrance (figure 11-13).

Cependant, pourquoi vouloir cacher l'un des rares critères d'accessibilité visible, qui témoigne de votre intérêt pour la question ? Vous faites des efforts, vous voulez sans doute qu'ils soient valorisés, non ? Plutôt que de mettre en place des gadgets de type loupe d'agrandissement dont on a vu qu'ils ne servaient à rien en termes d'accessibi-

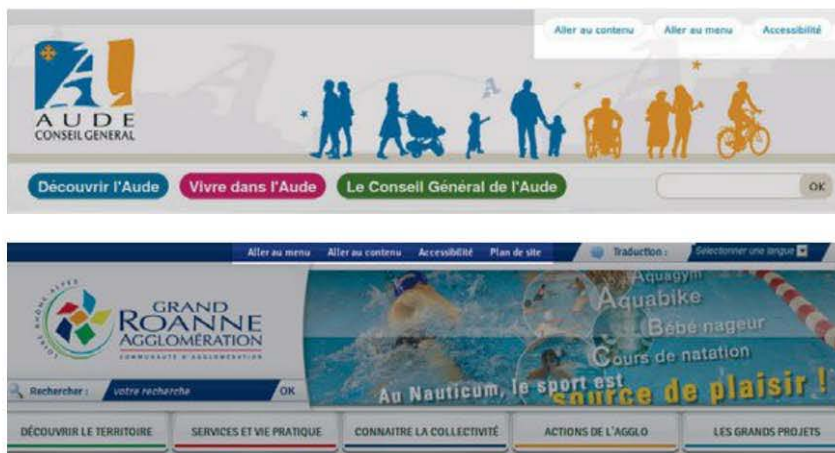


Figure 11-13 Sur le site de Légifrance, les liens d'accès rapide n'apparaissent que lorsqu'on appuie sur la touche Tab. (<http://www.legifrance.gouv.fr>)

lité, laissez ces liens visibles ! Vous pouvez même les mettre en valeur visuellement pour en faire un argument de communication, comme c'est le cas sur le site du Conseil général de l'Aude et sur celui de la communauté d'agglomération du Grand Roanne (figure 11-14).

Figure 11-14

Liens d'accès rapide mis en place respectivement sur le site du Conseil général de l'Aude (<http://www.aude.fr>) et sur celui de la communauté d'agglomération du Grand Roanne (<http://agglomero-grandroanne.fr>) : les liens sont bien visibles.



Notons que des liens d'accès rapide bien visibles, c'est aussi un gain de temps pour les utilisateurs de tablettes ou de smartphones, qui pourront ainsi naviguer plus rapidement sur un petit écran.

Attention, c'est un piège !

Nous terminerons par un critère extrêmement important : le critère 12.4, qui concerne les pièges au clavier. Un piège au clavier est défini par une situation qui vous empêche de naviguer dans la page. Par exemple, vous placez un script de diaporama en haut de page, mais vous ne pensez pas à le rendre accessible au clavier. Les personnes qui naviguent au clavier ne pourront jamais entrer dans votre page. Ils sont « piégés » dès l'entrée. Ce qui veut dire que même si vous respectez parfaitement de nombreux critères d'accessibilité au sein de la page, elle restera totalement inaccessible pour de nombreux utilisateurs.

Voici un exemple, tristement célèbre dans le milieu, du site de l'Élysée (figure 11-15).

Figure 11-15

Site de l'Élysée : quand on appuie sur la touche Tab pour entrer dans le site, on ne peut pas aller plus loin que le premier lien en JavaScript qui est supposé permettre la désactivation de l'animation. (<http://www.elysee.fr>)



Ce site, qui concerne pourtant le président de la République, président de « tous les Français », théoriquement, quelle que soit leur situation (handicap ou non) n'est pourtant prévu que pour une navigation à la souris. Essayez de naviguer au clavier en utilisant la touche **Tab** : vous resterez bloqué tout en haut, sur le lien en JavaScript « Animations : désactiver/activer » (qui, par ailleurs, ne fonctionne pas plus au clavier qu'à la souris...).

La consigne donnée par le critère 12.14 est simple : il faut permettre à l'utilisateur d'atteindre l'élément suivant ou précédent pouvant recevoir le focus avec la touche **Tab**. Et, à défaut, l'utilisateur doit être informé d'un mécanisme fonctionnel permettant d'atteindre au clavier l'élément suivant ou précédent pouvant recevoir le focus.

Un piège au clavier peut être extrêmement pénalisant en termes d'accessibilité. S'il est situé en début de page, comme dans cet exemple, il rend tout le site inaccessible au clavier, excluant de nombreuses personnes de l'accès aux contenus. S'il est placé en fin de page, c'est déjà moins gênant.

Dans tous les cas, veillez bien à ne pas piéger vos utilisateurs, il y a fort à parier qu'ils n'apprécieront pas...

Consultation

« Vérifier que l'utilisateur a le contrôle des procédés de rafraîchissement, des changements brusques de luminosité, des ouvertures de nouvelles fenêtres et des contenus en mouvement ou clignotant.

Indiquer lorsqu'un contenu s'ouvre dans une nouvelle fenêtre et donner des informations relatives à la consultation des fichiers en téléchargement. Ne pas faire dépendre l'accomplissement d'une tâche d'une limite

de temps sauf si elle est essentielle et s'assurer que les données saisies sont récupérées après une interruption de session authentifiée. S'assurer que les expressions inhabituelles et le jargon sont explicités. Proposer des versions accessibles ou rendre accessibles les documents en téléchargement. »

<http://ur1.ca/a0cff>⁴

La dernière des treize thématiques AccessiWeb s'intitule « Consultation ». Elle traite de l'accessibilité de tous les éléments, à l'écran ou en téléchargement, que nous sommes amenés à consulter.

Avez-vous le temps de consulter la page ? Des éléments perturbateurs comme des animations ne viennent-ils pas vous déconcentrer ? Le document que vous avez téléchargé est-il accessible ? Toutes ces questions, et d'autres encore, seront abordées dans cette thématique.

INFO Critères de la thématique « Consultation »

La thématique « Consultation » contient dix-sept critères, dont dix de niveau Bronze et sept de niveau Or. Nous n'aborderons pas ou ne ferons que survoler les critères Or suivants et vous laissons le soin de vous reporter au référentiel AccessiWeb pour en savoir plus.

- Critère 13.4 (amélioration du critère 13.1) :
▶ <http://ur1.ca/a4jkv>^a
- Critère 13.5 (interruption de session authentifiée) :
▶ <http://ur1.ca/a4jcx>^b
- Critère 13.14 (texte compréhensible pour un élève de fin de collège) :
▶ <http://ur1.ca/a4jlb>^c
- Critère 13.16 (amélioration du critère 13.15) :
▶ <http://ur1.ca/a4jli>^d

a. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-13-4

b. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-13-5

c. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-13-14

d. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#crit-13-16

Maîtriser son environnement : les limites de temps et l'ouverture de nouvelles fenêtres

La première des choses à faire consiste à permettre à vos utilisateurs de maîtriser leur environnement. En d'autres termes, il ne doit pas y avoir d'action déclenchée sans que l'utilisateur ne soit averti ou ne puisse agir pour contrôler l'action.

4. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#consultation

Rafraîchissement ou redirection automatique et limites de temps

Il peut arriver que vous visitiez une page et qu'un processus de redirection ou de rafraîchissement automatique soit prévu. Par exemple, vous effectuez un achat sur un site, et au moment de payer, vous basculez sur le site d'une banque. Vous effectuez le paiement, une page avec la confirmation s'affiche et au bout de quelques secondes, vous êtes redirigé vers le site sur lequel vous avez passé votre commande.

Il s'agit bien d'une redirection automatique. Selon la vitesse du processus et votre temps de réactivité, peut-être n'aurez-vous pas eu le temps de vérifier le détail de votre transaction ou d'imprimer la confirmation pour la conserver. Ceci est valable selon votre situation de handicap (certains handicaps mentaux ou cognitifs rendent la lecture plus lente et l'utilisation d'un lecteur d'écran rend également la lecture d'une page plus laborieuse, surtout quand elle n'est pas parfaitement accessible). Mais vous pouvez aussi être interrompu dans votre transaction par un coup de téléphone... Le temps de décrocher, vous êtes déjà redirigé sans avoir eu le temps d'imprimer.

Il est donc important de laisser le contrôle à l'utilisateur pour tout ce qui concerne les redirections ou rafraîchissements automatiques. À moins bien sûr qu'il ne s'agisse d'une redirection immédiate, et donc imperceptible pour l'utilisateur.

Laisser le contrôle à l'utilisateur signifie concrètement :

- lui permettre d'arrêter ou de relancer le processus ;
- lui offrir la possibilité d'augmenter de dix fois au moins la limite de temps entre deux rafraîchissements ou avant redirection ;
- l'avertir qu'un rafraîchissement ou une redirection va bientôt avoir lieu et lui laisser au moins vingt secondes pour augmenter la limite de temps ;
- la limite de temps est de vingt heures au moins, délai suffisamment long pour être considéré comme non handicapant.

Notons que le critère 13.4, de niveau Or, interdit tout simplement les limites de temps, sauf dans le cas où l'action se situe en temps réel ou quand la limite de temps est essentielle (c'est le cas, par exemple, pour un exercice devant se dérouler pendant une durée limitée).

Ouverture de nouvelles fenêtres

Les personnes qui ne peuvent pas détecter les changements ou se rendre compte qu'un changement a eu lieu peuvent être désorientées dans leur navigation. C'est le cas, par exemple, des personnes aveugles ou ayant une basse vision, qui peuvent avoir des difficultés à savoir quand un changement visuel s'est produit, comme l'ouverture de nouvelles fenêtres. Dans ce cas, prévenir les utilisateurs réduit les risques d'une confusion qui pourrait survenir quand ils découvrent que le bouton du navigateur pour revenir en arrière ne se comporte pas comme ils s'y attendent.

Par ailleurs, certaines personnes ayant une vision basse ou des troubles intellectuels ou de lecture peuvent avoir des difficultés à interpréter les signaux visuels. Des signaux supplémentaires les aidant à détecter ce changement leur seront bénéfiques.

L'ouverture de nouvelles fenêtres est encadrée par deux critères d'accessibilité, tous deux de niveau Bronze. La consigne est simple, il s'agit :

- de ne pas permettre l'ouverture de nouvelles fenêtres sans action de l'utilisateur (critère 13.3) ;
- d'avertir l'utilisateur si le lien s'ouvre dans une nouvelle fenêtre (critère 13.2).

Là encore, il s'agit de laisser le contrôle de sa consultation à l'internaute, pour qu'il ne soit pas perturbé.

Le critère 13.3 est très simple à comprendre et à mettre en pratique, contrairement au critère 13.2, qui pose question.

Les WCAG 2 précisent clairement que les changements doivent être annoncés pour permettre à tous (personnes non voyantes, mais aussi celles ayant une basse vision ou des troubles intellectuels ou cognitifs) d'être avertis. Pourtant, parmi les techniques autorisées par AccessiWeb, l'utilisation de l'attribut `title` sur les liens pour indiquer l'ouverture d'une nouvelle fenêtre est la plus fréquemment utilisée.

Or même si l'attribut `title` déclenche l'apparition d'une infobulle au survol, cela ne fonctionne pas au clavier. Pour peu que vous soyez obligé de naviguer au clavier et que vous ayez aussi un trouble cognitif, vous ne serez pas averti. Cette hypothèse est pourtant bien plus probable et moins théorique que le cas d'une désactivation des images avec les CSS activées. De même, le nombre de personnes concernées sera sans doute plus important que celui des personnes étant contraintes de naviguer sans JavaScript.

Selon nous, il manque ici un critère de pertinence. Le critère 13.2 indique bien qu'il faut avertir les utilisateurs (présence d'un avertissement), mais rien ne demande à ce que cet avertissement soit donné de façon pertinente.

Cette absence de critère (et donc d'obligation) concernant un avertissement visuel en complément d'un attribut `title` résulte d'une volonté de ne pas rendre l'accessibilité trop compliquée, avec des exigences trop contraignantes. Le critère 13.2 est en effet un critère de niveau Bronze.

Grâce à cette interprétation, vous pouvez prétendre à une labellisation en vous contentant d'ajouter un attribut `title` indiquant l'ouverture d'une nouvelle fenêtre, comme sur l'exemple de la figure 11-16 (page suivante).

La page « Accessibilité » du site de la MDPH du Jura (<http://www.mdp35.fr>) comporte en effet un lien externe qui s'ouvre dans une nouvelle fenêtre. Cette information est donnée grâce à l'attribut `title` du lien qui permet l'affichage d'une infobulle. Mais si nous naviguons au clavier, aucune information n'est donnée.



Figure 11-16 La page « Accessibilité » du site de la Maison départementale des personnes handicapées (MDPH) du Jura : au survol du lien, une infobulle indique que le lien s'ouvre dans une nouvelle fenêtre.

Alors, que faire pour que ces liens soient accessibles à tous ? La première des bonnes pratiques consiste à tout simplement éviter de contraindre l'utilisateur à ouvrir de nouvelles fenêtres. Laissez donc autant que possible les liens s'ouvrir normalement, les utilisateurs pourront choisir d'ouvrir la page dans une nouvelle fenêtre s'ils le souhaitent.

Malheureusement, la crainte, bien qu'infondée, de perdre un internaute qui quitte ainsi le site peut vous obliger à proposer malgré tout des liens s'ouvrant dans de nouvelles fenêtres, et ce n'est de toute façon pas interdit en termes d'accessibilité. Dans ce cas, le plus efficace est encore de mettre en toutes lettres dans l'intitulé du lien la mention « (nouvelle fenêtre) ». Efficace, c'est certain, mais pas forcément très agréable visuellement, nous en convenons, surtout quand la page dispose de nombreux liens de ce genre. La navigation peut perdre en fluidité avec des mentions « (nouvelle fenêtre) » partout.



Figure 11-17 Exemple d'une page du site de la WAI : on voit que plusieurs liens sont dotés d'une icône particulière afin d'indiquer qu'il s'agit de liens externes.

Une mention en bas à gauche rappelle la signification de cette icône. (<http://ur1.ca/abyhp>)

a. <http://www.w3.org/WAI/bcase/resources.html>

En alternative, vous pouvez utiliser un attribut `title` sur le lien, mais il faut également penser à ajouter une *indication visuelle supplémentaire* pour les internautes naviguant au clavier et ayant besoin de cette information.

Le site de la WAI utilise, par exemple, deux types d'icônes, dont l'une représente une flèche sortant d'un carré pour distinguer les liens externes (figure 11-17). Ceux-ci ne s'ouvrent pas dans une nouvelle fenêtre, mais cette icône renseigne sur la nature du lien.

Ainsi, si tous vos liens externes s'ouvrent dans une nouvelle fenêtre, à condition de le mentionner sur votre page d'aide, par exemple, l'internaute pourra le savoir au premier coup d'œil, et les internautes non voyants seront également informés via l'attribut `title` du lien.

Figure 11-18

Sur la page d'aide du site de la WAI, une section est consacrée à la signification des icônes associées aux liens.
(<http://ur1.ca/a8pad>^a)

Icons

The following icons are used on the WAI site:

-  icon indicates links to technical reports or other page formats without WAI site navigation. The alternative text description for this image is "technical report", "W3C Note", or "different format" depending on the type of pages it links to.
-  icon indicates links to off W3C WAI site.

a. <http://www.w3.org/WAI/sitehelp.html>

En résumé : éviter autant que possible de faire ouvrir vos liens dans de nouvelles fenêtres. Mais si vous devez le faire, prévenez par un moyen perceptible à la fois pour les utilisateurs de lecteurs d'écran et pour les personnes naviguant à la souris (attribut `title`), sans oublier les personnes naviguant au clavier (icône, par exemple).

ASTUCE Une infobulle accessible au clavier et à la souris

Hors une image, une autre possibilité pour signaler visuellement l'ouverture d'une nouvelle fenêtre au clavier consiste à rendre l'attribut `title` visible quand on tabule sur le lien.

Voici un exemple, parmi d'autres, combinant CSS2 et CSS3, ainsi que du JavaScript avec jQuery :

► <http://ur1.ca/a9vpm>^a

Notons que l'exemple proposé sur le site permet l'affichage de tout attribut `title`, y compris sur les balises `<abbr>` ou autres. Si vous voulez cibler les `title` sur les balises de lien `<a>`, vous devrez préciser le sélecteur ou ajouter une classe pour spécifier que vous ne visez que les liens ouvrant une nouvelle fenêtre.

a. http://www.websemantics.co.uk/resources/styled_accessible_tooltips/

Enfin, ne déclenchez jamais l'ouverture de nouvelles fenêtres sans action de la part de l'utilisateur.

Pour clore cette section, voyons un cas de mauvais usage de l'attribut `title` sur un lien pour indiquer l'ouverture d'une nouvelle fenêtre, alors que ce n'est pas le cas. Il s'agit du site du Fonds pour l'insertion des personnes handicapées dans la fonction publique (<http://www.fiphfp.fr>) (figure 11-19, page suivante).

Figure 11-19

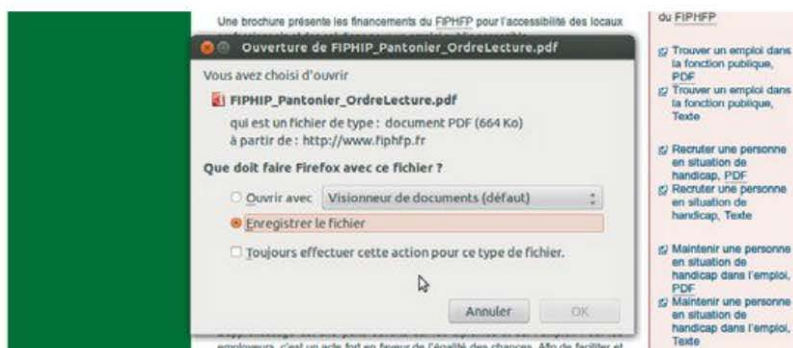
Au survol du lien pour télécharger un document, une infobulle indique que le lien s'ouvre dans une nouvelle fenêtre.



Le lien visible sur la figure permet de télécharger une brochure au format PDF. Cette action déclenche en effet l'ouverture d'une fenêtre pour donner le choix d'ouvrir ou d'enregistrer le document, par exemple (figure 11-20).

Figure 11-20

Quand on clique sur le lien, une fenêtre s'ouvre pour laisser le choix d'ouvrir ou d'enregistrer le document.



Cependant, cette fenêtre n'est pas une « nouvelle fenêtre » dans le sens où elle ne donne pas lieu à l'ouverture d'une nouvelle page web.

Attention, donc, à ne pas confondre ces deux notions. Il est également important de permettre à l'utilisateur de distinguer les liens ouvrant des documents, mais ce n'est pas l'objet des critères sur l'ouverture de nouvelles fenêtres. Voyons d'ailleurs ce qu'il en est des documents en téléchargement.

Documents en téléchargement

Deux grands principes régissent les documents en téléchargement : il faut qu'ils soient clairement identifiés (critère 13.6) et ils doivent être accessibles si nécessaire (critères 13.7 et 13.8).

Identifier les documents en téléchargement

Un document en téléchargement, quelle que soit sa nature (fichier texte, audio, vidéo...) peut être fourni par un lien ou par un formulaire. Or, pour prendre l'exemple du lien,

quand on l'active, il ne se comporte pas comme un lien ordinaire. Au lieu de diriger vers une page web, en restant dans le navigateur, il propose à l'internaute d'ouvrir ou d'enregistrer un document sur son ordinateur. Pour lire le fichier, l'internaute doit donc quitter le Web. C'est également le cas quand le document est téléchargé via un formulaire.

Toujours dans une optique visant à laisser à l'internaute le contrôle de ses actions, le critère 13.6 demande à ce que l'ouverture de documents soit distinguée des liens ou formulaires dirigeant vers une page web, en y associant certaines informations :

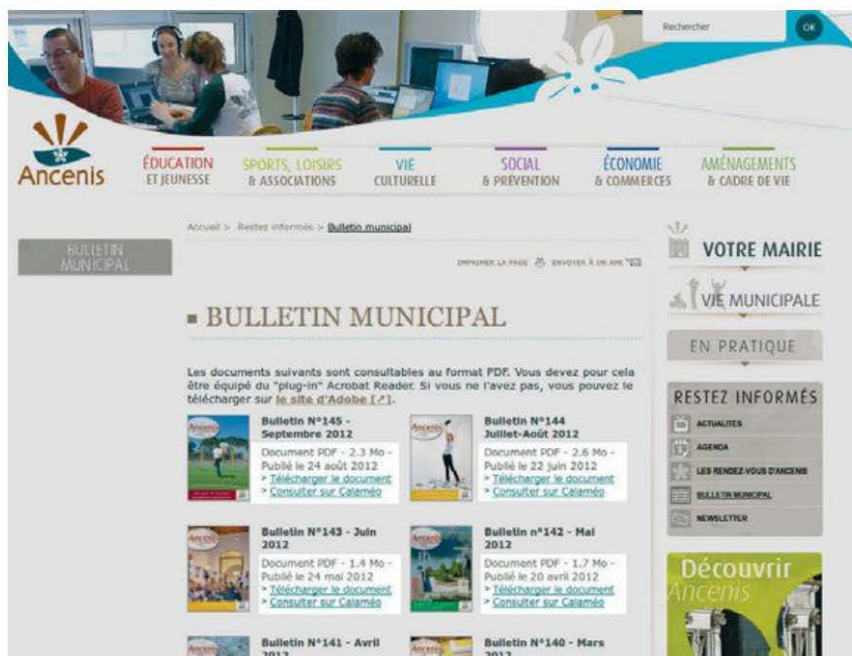
- le format du fichier (s'agit-il d'un fichier .pdf, .ogg, .odt... ?) ;
- le poids du document (en Ko, par exemple) ;
- la langue du document, si elle est différente de la langue de la page web à partir de laquelle le document peut être téléchargé. Par exemple, si la page est en français et que le document est également en français, il est inutile de préciser la langue du document. En revanche, si sur cette page en français, le document en téléchargement est en anglais, il est obligatoire de le mentionner.

L'essentiel, c'est que ces informations soient indiquées de façon accessible, bien sûr, et perceptible par tous (il ne faut donc pas qu'elles soient cachées).

Le site de la mairie d'Ancenis, par exemple, propose la liste des bulletins municipaux en téléchargement (figure 11-21). Des indications concernant le format et le poids de chaque document sont bien présentes.

Figure 11-21

Un texte accompagne chaque document en précisant le format (PDF) et le poids. (<http://ur1.ca/a91a5a>)

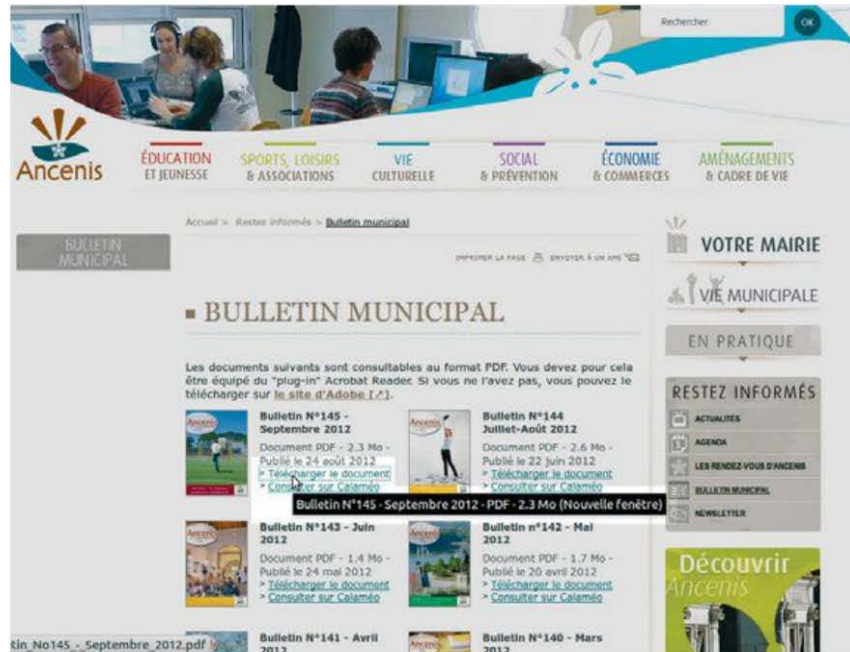


a. <http://www.ancenis.fr/-Bulletin-municipal-.html>

Ces informations, telles qu'elles sont données, sont suffisantes pour valider le critère 13.6. Mais notons que le site propose en plus un attribut `title` sur le lien pour le rendre encore plus explicite (figure 11-22).

Figure 11-22

Une infobulle reprend le nom du fichier à télécharger, son format et son poids, pour le rendre encore plus explicite.



Cette bonne pratique est facile à mettre en œuvre et rendra de grands services à vos utilisateurs non voyants pour qui le lien sera plus explicite hors contexte.

Remarquons enfin qu'il existe un cas particulier où le critère ne s'applique pas totalement. Quand il s'agit de la création d'un PDF à la volée, pour l'émission d'une facture, par exemple, il n'est pas possible d'anticiper le poids du document. Par contre, le format et la langue (si nécessaire) restent exigibles.

Des documents bureautiques accessibles

Combien de fois avons-nous vu des organismes faire des efforts pour rendre leur site plus accessible, puis quelques jours plus tard, publier des pages web, toujours accessibles, mais dont le contenu indique par exemple : « Grande foire annuelle ! Pour connaître les détails, téléchargez le programme. » Toute l'information est donc contenue dans un fichier en téléchargement, très rarement accessible. Or, ce qui compte, c'est que l'information soit accessible, pas que le site soit accessible alors que les contenus ne le sont pas.

Ainsi, en passant évidemment sur l'hypothèse où votre page n'est pas accessible, il ne reste que deux alternatives :

- la page web est accessible et contient toute l'information utile, le document en téléchargement n'est donc qu'un plus ; il n'est pas nécessaire qu'il soit accessible ;
- la page web est accessible mais l'information est donnée dans un document en téléchargement ; dans ce cas, le document en téléchargement doit également être accessible.

Les règles pour rendre un document bureautique accessible sont les mêmes que pour le Web : dissociation du fond (structure/contenus) et de la forme grâce à l'utilisation des feuilles de styles, structuration du document avec des titres et des listes, alternatives aux images, etc.

Nous n'entrerons pas dans le détail de la réalisation de documents bureautiques accessibles, mais s'il s'agit d'un point que vous ne maîtrisez pas, il ne vous reste que deux options :

- vous former auprès d'un professionnel compétent ;
- reprendre toutes les informations données dans votre document sur la page web elle-même afin de fournir une alternative accessible.

Cette seconde solution est à la fois la plus simple et la plus efficace. Elle vous permettra de valider le critère 13.7, à condition que l'on retrouve la même information dans la version accessible (critère 13.8).

Le glossaire AccessiWeb précise que les formats de documents dont la compatibilité est reconnue sont :

- Microsoft Office (Word 2003, OOXML) ;
- Open Office Org (ODF) ;
- Adobe PDF ;
- Daisy.

Notons que le format texte (.txt) ne peut pas être utilisé pour produire une version accessible d'un document en téléchargement, puisqu'il ne peut pas être structuré.

À SAVOIR Liste des critères d'accessibilité pour les documents bureautiques

Vous trouverez en téléchargement une liste des critères d'accessibilité pour les documents bureautiques en annexe du référentiel AccessiWeb 2.2 :

► <http://ur1.ca/9yks3>^a

a. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#annexe

Des textes clairs et explicites

Attention au jargon !

Le critère 13.9 concerne les expressions inhabituelles, les expressions idiomatiques ou le jargon. Nous avons tenu à vous indiquer l'existence de ce critère de niveau Or, parce qu'il peut être utile sur des sites grand public. Il n'est toutefois classé qu'au niveau Or, car expliciter du jargon sur des sites spécialisés n'aurait pas de sens. En effet, qui dit site spécialisé, dit public de spécialistes. Faire du critère 13.9 un critère obligatoire pour tous les sites reviendrait à interdire certains sites.

Toutefois, il peut être bon de se donner pour principe de respecter ce critère quand on entend toucher le grand public.

Dans ce cas, il vous faudra ajouter une définition de l'expression :

- via l'utilisation de la balise `<dfn>` dans le contexte adjacent de l'expression ;
- via une liste de définition (balises `<dl>`, `<dt>` et `<dd>`) ;
- en donnant une définition directement dans la page ;
- en mettant sur l'expression un lien conduisant vers sa définition, dans un glossaire, par exemple.

À SAVOIR La balise `<dfn>`

La balise de définition `<dfn>` n'est pas souvent utilisée. Elle vise à encadrer une expression qui se voit définie dans le même paragraphe, par exemple. Ainsi, on pourrait écrire :

```
<p>Le <dfn>référentiel AccessiWeb</dfn> est un outil permettant d'évaluer le degré de conformité d'une page web par rapport aux règles internationales d'accessibilité des contenus web conçues par le W3C.</p>
```

Pour tout savoir sur cette balise dans la spécification HTML5, rendez-vous à l'adresse suivante :

► <http://ur1.ca/a918r^a>

a. <http://dev.w3.org/html5/spec/the-dfn-element.html>

Contenus cryptiques (émoticônes, art ASCII)

Si le critère 13.9 était facultatif, car de niveau Or, le critère 13.10 est, quant à lui, de niveau Bronze et peut donc être validé. Il concerne les contenus qu'on ne peut pas comprendre à la lecture par une synthèse vocale. Pour sa part, le critère 13.11 cherche à vérifier si l'alternative donnée en critère 13.10 est pertinente. Ces critères concernent notamment les dessins réalisés sous forme de signes textuels (art ASCII, smileys, etc.).

Dans ce cas, il faut prévoir d'en donner la signification soit via un attribut `title`, soit en donnant la définition en toutes lettres dans le contexte adjacent.

Par exemple :

```
<p>Salut toi ! :) [sourire]</p>
```

Ou encore :

```
<p>Salut toi ! <span title="sourire">:)</span></p>
```

Une solution bien plus efficace est de remplacer les images réalisées avec des signes textuels par une véritable image avec l'attribut `alt` qui convient. Mais cela relève alors de la thématique 1 sur les images (voir le chapitre 6).

Prononciation

Ici aussi, nous retrouvons un critère de niveau Or. Le critère 13.12 demande que chaque mot dont le sens ne peut être compris sans en connaître la prononciation soit explicité, soit en indiquant la prononciation phonétique dans le texte, soit via un lien.

Par exemple, il faudrait expliciter ce texte :

```
<p>Le président chinois (<span lang="cn"></span>) devrait bientôt  
laisser sa place à son actuel vice-président (<span lang="cn"></  
span>).</p>
```

Comme ceci :

```
<p>Le président chinois (<span lang="cn"></span> Hu Jintao) devrait  
bientôt laisser sa place à son actuel vice-président (<span lang="cn"></  
span> Xi Jinping).</p>
```

Nous voyons à travers cet exemple que même s'il s'agit d'un critère de niveau Or, c'est un réflexe que nous devrions avoir pour tous les utilisateurs.

À NOTER Quel est votre niveau d'études ?

Nous n'aborderons pas en détail le critère 13.14, mais il est intéressant de savoir qu'une des recommandations demande que les textes soient compréhensibles pour un élève arrivé à la fin du premier cycle de l'enseignement secondaire. Bref, un élève de troisième doit pouvoir comprendre ce que vous écrivez.

L'objectif est de prendre en compte le handicap mental. Le problème principal qui se pose ici est : comment évaluer ce qu'un élève de troisième peut comprendre ou non ? Par ailleurs, plusieurs problèmes se posent : il n'est pas toujours possible de simplifier un texte sans risquer d'en modifier le sens (cas des documents juridiques, par exemple), un site n'est pas toujours fait pour être grand public et le simplifier peut être compliqué, etc. Mais ce critère est de niveau Or, donc à but d'amélioration seulement.

Effet de flash, contenus en mouvement et clignotants

Quand un contenu est en mouvement ou qu'il clignote, il va inévitablement attirer le regard et perturber l'attention de la personne qui consulte la page. Il faut donc que l'utilisateur puisse en avoir le contrôle (critère 13.17). Par ailleurs, les flashes lumineux peuvent provoquer des crises d'épilepsie, il est donc important de les éviter (critère 13.15). Ces deux critères sont de niveau Bronze.

Il est assez simple de laisser le contrôle à l'utilisateur sur les contenus en mouvement ou clignotants :

- soit la durée du mouvement/clignotement est inférieure ou égale à cinq secondes, auquel cas il n'y a rien à faire ;
- soit la durée est supérieure à cinq secondes (cas d'un diaporama défilant sans cesse) et l'utilisateur doit alors pouvoir :
 - arrêter et relancer le mouvement/clignotement ;
 - masquer et afficher le contenu en mouvement ou clignotant.

Notez bien qu'il faut pouvoir arrêter, mais également relancer.

Par ailleurs, si l'utilisateur peut afficher la totalité de l'information sans le mouvement/clignotement, le critère est également validé.

Le critère 13.15 est plus complexe à évaluer, mais très important.

Pour éviter de provoquer des crises chez des personnes atteintes d'épilepsie photosensible, il convient de s'assurer que les flashes ne sont pas trop rapides (moins de trois par seconde) ; sinon, la surface à l'écran doit être réduite, c'est-à-dire correspondre à 25 % du champ visuel (sur un écran d'une résolution moyenne de 1 024 × 768 pixels, la surface totale cumulée des effets doit être inférieure ou égale à 21 824 pixels).

Concrètement, pour calculer la surface concernant les images, on trouve les dimensions dans les propriétés de l'objet. Lorsque ce sont des objets Flash, les dimensions sont généralement fixées en CSS ; il suffit alors de récupérer l'information, avec l'extension Firebug pour Firefox, par exemple. Sinon, il faut calculer la taille de l'objet avec une règle à pixels telle que MeasureIt pour Firefox ou un dispositif équivalent. Il convient ensuite de multiplier la longueur par la largeur et de vérifier que le total fait moins de 21 824 pixels.

Notons qu'un objet d'une surface cumulée de 21 824 pixels correspond à une surface carrée de 148 pixels de côté environ ($\sqrt{21\,824}$).

OUTILS MeasureIt : calculer une dimension en pixels sur une page web

Pour calculer une dimension d'un objet en pixels sur une page web, vous pouvez utiliser l'extension MeasureIt pour Firefox :

► <http://ur1.ca/4mfjg>^a

Une fois installée, vous pouvez activer l'extension en l'intégrant dans votre barre des modules (clic droit sur la barre des modules dans Firefox, puis clic sur *Personnaliser*). Une règle jaune apparaît. En cliquant dessus, vous pouvez prendre des mesures à l'écran, les dimensions en pixels s'affichent en temps réel. Quand vous avez terminé, cliquez à nouveau sur la règle pour désactiver l'outil.

a. <https://addons.mozilla.org/fr/firefox/addon/measureit/>

En concluant cette thématique 13 dédiée à la consultation, nous arrivons au terme de l'étude du référentiel AccessiWeb 2.2.

Nous avons vu que ce référentiel était basé sur du HTML4. Qu'en est-il alors de l'accessibilité de HTML5 ? C'est ce que nous allons voir dans le chapitre qui suit.

HTML5, ARIA et accessibilité

« L'innovation systématique requiert la volonté de considérer le changement comme une opportunité. »

Peter Drucker¹



1. Auteur sur le management d'entreprise

Nous venons de voir en détail le référentiel AccessiWeb 2.2. Les règles générales d'accessibilité qui sont décrites restent valables quelle que soit la technologie utilisée. Mais contrairement aux WCAG 2 qui ne reposent sur aucune technologie, AccessiWeb 2.2 a été conçu pour aider les développeurs en déterminant clairement ce qui est « compatible avec l'accessibilité » et ce qui ne l'est pas.

De cette analyse résulte un référentiel qui ne repose que sur des versions de HTML antérieures à HTML5.

Revenons sur la notion de « compatible avec l'accessibilité ». Cela signifie que la technologie utilisée doit fonctionner avec les technologies d'assistance et les fonctionnalités d'accessibilité des systèmes d'exploitation, des navigateurs ainsi que d'autres agents utilisateurs. Ceci de manière à n'exclure personne.

Or, même parmi les versions les plus récentes des technologies d'assistance et des navigateurs, l'implémentation des nouveautés de HTML5 n'est encore que partielle. Et les utilisateurs n'utilisent pas forcément ses versions les plus récentes, que ce soit pour des raisons économiques ou de formation.

RESSOURCES Évaluation de l'accessibilité de HTML5, par navigateur

Pour en savoir plus sur la prise en compte des nouvelles balises HTML5 par les navigateurs et leur accessibilité, vous pouvez vous référer au site développé par The Paciello Group (en anglais), et dont la dernière mise à jour au moment où nous écrivons ces lignes date de février 2012.

Les tests ne portent que sur les systèmes d'exploitation Windows et Mac OS, et l'on constate que la prise en compte est très mauvaise, à l'exception notable de Firefox sous Windows qui se détache nettement des autres.

► <http://html5accessibility.com>

Et pour savoir précisément quelle balise est implémentée dans quel navigateur, vous pouvez consulter par exemple le site suivant :

► <http://caniuse.com>

À l'heure où nous écrivons ces lignes, nous pouvons donc comprendre la décision d'AccessiWeb 2.2 de rester basé sur des versions plus stabilisées de HTML. Notons que le RGAA repose lui aussi sur les versions antérieures à HTML5.

À SAVOIR Ces balises HTML5 non reconnues...

Que deviennent donc ces nouveaux éléments HTML5 non implémentés dans les navigateurs ? Dans la plupart des cas, ils sont tout simplement ignorés. Le texte entre les balises est quant à lui restitué malgré tout, mais brut, sans l'apport des balises sémantiques.

Et qu'en est-il des technologies d'assistance ? Elles interprètent ce qui est fourni par le navigateur, donc si l'information n'est pas donnée par le navigateur, elle ne peut pas la restituer. Et si l'information est donnée, mais que la technologie d'assistance ne la reconnaît pas, elle-même va se comporter comme le navigateur en transmettant le texte entre les balises, mais sans l'apport sémantique des balises inconnues d'elle.

L'autre raison qui fait qu'AccessiWeb 2.2 n'a pas pris en compte HTML5, c'est justement que de nouveaux éléments et de nouvelles façons de coder (la hiérarchie des titres, notamment) doivent être pris en compte, ce qui suppose une réécriture de plusieurs tests et critères. La décision de BrailleNet a donc consisté à faire une mise à jour du référentiel AccessiWeb 2.2 toujours basé sur HTML4, mais plutôt que de faire une version 2.3, l'objectif est de rédiger un autre référentiel AccessiWeb, basé lui sur HTML5. À terme, il y en aurait donc deux qui coexisteraient.

DÉBAT HTML5, work in progress... ou pas !

Dans un article de Numerama publié le 23 juillet 2012, l'auteur indique : « Le 19 juillet dernier, Ian Hickson a publié un message dans la liste de diffusion du W3C (*World Wide Web Consortium*) concernant le HTML5. L'homme y annonce qu'un fork du langage venait d'être effectué, créant ainsi deux versions parallèles du langage de référence de la création des sites et applications web. ».

La raison de cette scission du langage ? Une divergence de vues sur la gestion du projet HTML5. Alors que le W3C, conforme à sa tradition, souhaite passer par des étapes de revues privées et publiques, avec des montées de versions selon les retours et un passage progressif vers une version stable de HTML5 (ce qui prendra encore de nombreuses années), le groupe de travail WHATWG, composé notamment d'éditeurs de navigateurs et à l'initiative de HTML5, préfère adopter une version « en mouvement », avec une adoption progressive et continue des changements techniques pour un langage qui ne serait jamais finalisé car les évolutions peuvent se poursuivre indéfiniment.

On voit là s'opposer deux méthodes de gestion de projet : une méthode traditionnelle et une méthode plus « agile ».

Quel est la meilleure option pour l'accessibilité ?

Difficile à dire. La version stabilisée rend beaucoup plus facile la définition de ce qui est ou non compatible avec l'accessibilité. On sait sur quoi s'appuyer, chaque interlocuteur (développeur de sites, de navigateurs, de technologies d'assistance) peut s'appuyer sur du solide, du stabilisé. L'inconvénient, c'est que ce processus prend énormément de temps, beaucoup trop de temps par rapport aux avancées techniques, elles-mêmes porteuses d'améliorations pour l'accessibilité.

De son côté, la version « agile » est plus fluide, permet des avancées à petits pas, mais la contrepartie négative est que tous les utilisateurs, selon leurs moyens techniques, ne sont plus à égalité. Surtout en matière d'accessibilité. Cela rend le travail des développeurs de sites bien plus difficile par ailleurs, car il faut pouvoir apporter des alternatives (*fallback*) pour les fonctionnalités pas encore implémentées partout. Bref, il n'y a pas de réponse simple à ce problème, et nous n'avons pas la prétention de trancher un débat en cours. Chaque position comporte son lot d'effets positifs et négatifs pour l'accessibilité. À suivre donc...

Pour lire l'article de Numerama :

► <http://ur1.ca/9tknd^a>

Vous pouvez également lire un article intéressant faisant le point sur le sujet, sur le site de HTeuMeuLeu, intitulé le « W3C, le WHATWG et le HTML5 » :

► <http://ur1.ca/a0ef0^b>

a. <http://www.numerama.com/magazine/23240-html5-deux-versions-pour-diviser-les-developpeurs.html>

b. <http://www.hteumeuleu.fr/le-w3c-le-whatwg-et-html5/>

Pourtant, c'est un fait, HTML5 existe déjà sur le Web. De plus en plus de développeurs conçoivent leurs sites avec HTML5. De nombreux logiciels de gestion de contenus (CMS) proposent désormais des modèles de mise en page (gabarits HTML ou *templates*) codés par défaut en HTML5. Et comme toujours, les évolutions technologiques étant plus rapides que les évolutions normatives, il y a fort à parier que l'accessibilité de HTML5 sera une réalité pour de nombreux utilisateurs avant qu'un référentiel dédié ne voit le jour. Il est donc important de faire le point sur l'accessibilité de cette technologie.

Le contexte étant posé, voyons quelques-unes des nouveautés de HTML5 et leurs conséquences en matière d'accessibilité.

Chouette, encore plus de sémantique !

Le premier constat que nous pouvons faire quand nous observons les nouvelles balises introduites en HTML5, c'est l'amélioration apportée sur la structure sémantique de la page. De nombreux éléments qui manquaient manifestement en HTML4 ont été créés pour définir avec plus de précision la nature des contenus.

Cela est vraiment une excellente nouvelle pour l'accessibilité ! Car les technologies d'assistance vont pouvoir récupérer cette information. Enfin, à terme, quand ce sera suffisamment implémenté et donc « compatible avec l'accessibilité »...

Une meilleure structuration de la page

Les intégrateurs consciencieux, lorsqu'ils codent une page en HTML4 ou en XHTML 1.x, s'appliquent toujours à structurer leurs pages web avec des balises `<div>` (non sémantiques) auxquelles ils ajoutent un attribut `id` pour délimiter les zones de contenu importantes.

Les zones que l'on retrouve la plupart du temps sont les suivantes : en-tête, contenu principal, menu de navigation, contenu annexe, article (pour un blog, par exemple) et enfin, le pied de page.

Ce qui donne quelque chose de cet ordre-là :

```
<div id="header">
<h1>Titre de ma page</h1>
<h2>Sous-titre de ma page</h2>
</div>
<div id="content">
<div id="article1">
<h3>Titre de l'article</h3>
```

```

<p>Un premier paragraphe</p>
<p>Un second paragraphe</p>
</div><!--fin article1-->
<div id="article2">
<h3>Titre de l'article</h3>
<p>Un premier paragraphe</p>
<p>Un second paragraphe</p>
</div><!--fin article2-->
</div><!--fin content-->
<div id="sidebar">
<h3>Édito du jour</h3>
<p>Un premier paragraphe</p>
<p>Un second paragraphe</p>
<h3>Liens utiles</h3>
<ul>
<li><a href="/1">Lien 1</a></li>
<li><a href="/2">Lien 2</a></li>
<li><a href="/3">Lien 3</a></li>
</ul>
</div><!--fin de sidebar-->
<div id="footer">
<p>Site réalisé par Môaa...</p>
</div><!--fin de footer-->

```

Ce genre de structure vous rappelle quelque chose ? Rien de surprenant, c'est en général ce qu'on retrouve en HTML4 (ou XHTML 1).

Actant de ce phénomène, HTML5 a pris en compte cette structuration en ajoutant de nouvelles balises donnant le sens que les balises `<div>` ne peuvent véhiculer. Ainsi, au niveau de la page, nous trouvons les nouvelles balises suivantes : `<article>`, `<aside>`, `<section>`, `<nav>`, `<header>`, `<footer>`, et même `<hgroup>` pour regrouper des titres avec leurs sous-titres.

Si nous reprenons le bout de code que nous avons pris en exemple ci-dessus, voici donc ce qu'il donnerait en HTML5 :

```

<header>
<hgroup>
<h1>Titre de ma page</h1>
<h2>Sous-titre de ma page</h2>
</hgroup>
</header>
<section id="content">
<article>
<h3>Titre de l'article</h3>
<p>Un premier paragraphe</p>
<p>Un second paragraphe</p>
</article>

```

```
<article>
<h3>Titre de l'article</h3>
<p>Un premier paragraphe</p>
<p>Un second paragraphe</p>
</article>
</section><!--fin content-->
<aside>
<section>
<h3>Édito du jour</h3>
<p>Un premier paragraphe</p>
<p>Un second paragraphe</p>
</section>
<section>
<h3>Liens utiles</h3>
<ul>
<li><a href="/1">Lien 1</a></li>
<li><a href="/2">Lien 2</a></li>
<li><a href="/3">Lien 3</a></li>
</ul>
</section>
</aside>
<footer>
<p>Site réalisé par Môaa...</p>
</footer>
```

RESSOURCE Tout savoir sur HTML5

Pour tout savoir sur HTML5, nous vous invitons à lire à ce sujet le livre très complet de Rodolphe Rimelé, que nous vous avons déjà indiqué en introduction :

📖 Rodolphe Rimelé, *HTML5 : une référence pour le développeur web*, Eyrolles, 2011

L'intérêt pour l'utilisateur en situation de handicap ? C'est la définition programmatique de la nature des éléments.

Alors que la balise `<div>` est une balise neutre, sans contenu sémantique, visant seulement à définir des blocs de contenus pour faire de la mise en page en CSS, les balises que nous venons de voir apportent une information. Et cette information pourra être interprétée par les technologies d'assistance, comme les lecteurs d'écran, apportant un niveau de précision sur la structure de la page qu'il était impossible de donner avant cela.

Le gain pour une personne aveugle est donc évident, puisque le lecteur d'écran s'appuie sur ces éléments sémantiques pour restituer l'information. Mais on peut également imaginer un gain de temps pour des personnes en situation de handicap moteur, pouvant alors naviguer non seulement de titre en titre, mais également d'article en article, ou de section en section, etc. Enfin, si une technologie d'assis-

tance souhaitait fournir une réorganisation de la page simplifiée pour certains handicaps cognitifs, ce serait désormais possible.

HTML5 offre donc, avec ces nouvelles balises de structure, l'opportunité pour les technologies d'assistance de fournir de nouvelles fonctionnalités pour toujours plus d'accessibilité !

À SAVOIR `<time>`, `<bdo>`, `<bdi>`...

Nous venons de voir des balises sémantiques de type bloc, mais HTML5 introduit également, sur le même principe, plus de sémantique via des balises de type « en ligne » comme `<time>`, `<bdo>` ou `<bdi>`.

Nouvel algorithme de titrage

Cette organisation de la page via de nouvelles balises de structure a une conséquence directe sur la façon d'organiser la hiérarchie des titres.

Un nouvel algorithme a en effet été mis en place pour permettre une double approche.

- **Une vue par page** : la page doit être structurée de façon logique, ne pouvant passer de `<h1>` à `<h3>` sans passer par `<h2>` par exemple. C'était déjà le cas en HTML4. Et c'est ce que cherche à vérifier le critère AccessiWeb 9.1.
- **Une vue par bloc** : en HTML5, la page est composée de différentes entités indépendantes. Ainsi, un article (balise `<article>`) peut être traité de façon isolée et comporter une hiérarchie de titres commençant par `<h1>`. La suite de la page peut poursuivre par un `<h4>` sans que cela constitue une erreur en HTML5, même si le titre précédent qui se trouvait dans l'article était un `<h2>`, sans `<h3>` intermédiaire.

Le nombre de balises `<h1>` tend donc à se multiplier en HTML5. Cela permet une organisation de la page suivant une pensée plus riche et plus complexe, mais rend également difficile la représentation mentale d'une telle structure par une personne ne voyant pas la page.

Pour bien faire, les technologies d'assistance devraient être en mesure de rendre cette double vue (par page et par bloc) en réorganisant la hiérarchie des titres selon le mode utilisé. Ainsi, si un utilisateur aveugle demande l'affichage des titres, selon s'il demande les titres de la page ou les titres de l'article, le lecteur d'écran devrait fournir une information adaptée.

Pour l'instant, ce nouvel algorithme n'est pris en charge par aucune technologie d'assistance. Ce n'est donc pas « compatible avec l'accessibilité ». Cela signifie que si vous utilisez HTML5 et hiérarchisez vos titres selon les règles de ce langage, les technologies d'assistance vont continuer à les restituer sur le modèle simpliste et linéaire de HTML4, ce qui risque de ne pas être très clair pour vos utilisateurs.

Des légendes pour les images

Dans les exemples que nous avons pris pour la thématique « Images » (voir le chapitre 6), nous avons vu que sur les sites d'information, les images, qu'elles soient ou non porteuses d'information, étaient systématiquement accompagnées d'un texte faisant office de légende. Or, ce texte était bien relié visuellement à l'image, mais rien ne permettait de lier la légende à l'image d'un point de vue informatique.

HTML5 comble cette lacune en introduisant les balises `<figure>` et `<figcaption>`.

À SAVOIR `<figure>/<figcaption>`, ce n'est pas que pour les images !

Rappelons que tout élément visuel apporté par HTML peut comporter une légende et donc utiliser le couple de balises `<figure>` et `<figcaption>`. Cela peut donc concerner les images, mais aussi les vidéos, un dessin avec la balise `<canvas>`, etc.

Notre propos concernant l'utilisation de l'attribut `alt`, nous ne nous attarderons ici que sur l'exemple des images.

Voici un exemple d'utilisation de ces balises, reposant sur l'article du *Point* sur le Tsunami évoqué plus haut :

```
<figure>

<figcaption>
<p>Une vague emportant tout sur son passage est remontée dans les
terres</p>
</figcaption>
</figure>
```

Nous pouvons, grâce à `<figcaption>`, donner une légende liée à l'image. Mais nous pouvons aussi mettre un attribut `alt` vide puisque l'image n'apporte pas plus d'informations que le texte qui l'accompagne.

On pourrait donc penser que le problème que nous soulevions dans l'encart page 93 serait résolu grâce à HTML5, quand il sera suffisamment implémenté et donc « compatible avec l'accessibilité ». En réalité, les choses sont plus complexes. Tout dépend de la façon dont les technologies d'assistance vont traiter ces nouvelles balises. Pour le moment, une image avec un `alt` vide est ignorée. Quid de sa légende alors ?

L'idéal serait sans doute que les lecteurs d'écran offrent aux utilisateurs aveugles la possibilité de paramétrer leur logiciel pour restituer ou non les légendes associées aux images, que `alt` soit renseigné ou non. Du point de vue de l'utilisateur en tous cas, l'association entre une balise `<img />` avec un attribut `alt` vide et une légende donnée par la balise `<figcaption>` ne pose pas de problème particulier.

Pour le moment donc, et en attendant que la question soit officiellement tranchée, nous vous recommandons de traiter les images en fonction des informations dont elles sont porteuses ou non. N'hésitez pas non plus à utiliser ces nouvelles balises qui, si elles ne sont pas restituées, ne posent pas plus de problème de compréhension, mais apportent un bénéfice sémantique si elles sont prises en charge.

DÉBAT Supprimer l'attribut alt ?

Un vif débat fit rage pendant un temps sur les listes de discussion du W3C, certains souhaitant la disparition pure et simple de l'attribut `alt` au bénéfice des nouvelles balises `<figure>` et `<figcaption>`.

Finalement, l'argument de la rétrocompatibilité l'a emporté, mais la question du traitement de l'attribut `alt` quand l'image reçoit une légende via les balises `<figure>` et `<figcaption>` est loin d'être tranchée.

Le conseil que nous vous donnons ci-dessus n'est donc pas une position officielle, et notre point de vue lui-même peut être amené à évoluer selon l'avancée du débat en cours.

Des champs de formulaire mieux identifiés

Concernant les formulaires, plusieurs nouveaux attributs sont introduits en HTML5.

Jusqu'ici, quand nous utilisions JavaScript ou PHP pour rendre un champ obligatoire, par exemple, rien ne permettait de distinguer sémantiquement un champ obligatoire d'un champ qui ne l'était pas dans le code HTML. HTML5 permet de le faire directement en ajoutant des attributs spécifiques sur les balises de champs de formulaire `<input>`. Il n'y a pas moins de onze nouveaux attributs introduits : `placeholder`, `required`, `pattern`, `autocomplete`, `autofocus`, `list`, `multiple`, `dirname`, `min`, `max`, `step`.

L'objet de ce livre n'étant pas HTML5, nous n'entrerons pas dans le détail de tous ces nouveaux éléments. En revanche, il est important de constater que l'ajout de ces attributs directement dans le code HTML est une avancée notable pour l'accessibilité. Cela va permettre une meilleure identification de ces éléments.

Bref, encore un point positif de HTML5 pour l'accessibilité, à terme.

Le multimédia devient du Web

Internet est un vaste réseau de réseaux, chacun fonctionnant avec son propre protocole : le Web, bien sûr, avec son protocole TCP/IP, mais aussi la messagerie ou les médias en streaming reposant sur le protocole RTP...

Le multimédia, ce n'est donc pas seulement du Web. Le seul moyen d'embarquer un objet multimédia jusqu'ici était donc de passer par un objet non web créé en Flash très souvent, et de l'embarquer sur la page via des balises dédiées à l'ajout d'objets extérieurs à la page. Cela signifie qu'une personne aveugle souhaitant entrer dans un objet multimédia devait pouvoir l'identifier et décider d'y entrer ou non (ce qui nécessite une manipulation).

Avec HTML5, deux nouvelles balises viennent révolutionner le multimédia sur le Web, puisque la présentation de ces médias peut être faite nativement par le navigateur en HTML grâce à `<video>` et `<audio>`.

La prise en compte par les navigateurs et les technologies d'assistance sera donc native, comme pour tout élément HTML, sans avoir besoin d'ajouter des extensions propriétaires au navigateur. Cela implique également que l'objet multimédia fera désormais partie intégrante de la page web.

Le lecteur natif embarqué par les balises `<audio>` et `<video>`, et donc l'accessibilité des contrôles du média appliqués par l'attribut `controls`, sont du ressort des navigateurs qui permettront, à terme, de mettre très facilement en ligne du multimédia accessible sans beaucoup de travail pour les concepteurs de sites.

À NOTER Personnaliser ses contrôles en JavaScript

Il est possible de surcharger ces contrôles en JavaScript pour les personnaliser. Pour l'heure, c'est même une pratique recommandée pour pallier les lacunes des navigateurs dont l'implémentation de `controls` n'est pas encore accessible partout.

Par ailleurs, il est bien prévu une balise pour le sous-titrage, bien que peu implémentée à ce jour. Il s'agit de la balise `<track>`.

Nous regrettons malheureusement que rien n'ait été prévu pour l'audiodescription. Il sera donc toujours nécessaire de faire deux versions du média (avec et sans audiodescription), alors qu'il aurait été plus pertinent de permettre, comme pour les sous-titres, l'ajout d'une bande sonore correspondant à l'audiodescription, indépendante du média (non incrustée) et que l'on puisse charger à la demande. On pourrait même imaginer un simple fichier texte à charger, qui serait vocalisé de façon synchronisée, de sorte que la synthèse vocale puisse faire office d'audiodescription.

Enfin, il vous faudra être vigilant dans votre utilisation de l'attribut `autoplay`, qui lance la lecture du média automatiquement. Évidemment, les critères 4.18 (pour le son) et 13.17 (pour les images en mouvement) autorisent le lancement automatique d'un média à condition qu'il soit contrôlable par l'utilisateur, et donc qu'il puisse l'arrêter. Mais cela risque fort de perturber vos utilisateurs. À éviter quand vous le pouvez.

Si vous deviez vraiment lancer automatiquement un média, pensez à placer un contrôle au tout début de votre page, comme premier élément, pour arrêter le média. Cela évitera aux internautes aveugles de chercher, avec une synthèse vocale, le contrôle dans l'objet multimédia. Cette situation provoque en effet une ambiance sonore indescriptible, entre le média lui-même actif qui fait du bruit et la voix de la synthèse vocale...

À SAVOIR `<embed>` devient standard

La balise `<embed>`, qui était considérée comme une balise propriétaire non standard en HTML4, devient officielle en HTML5.

Peut-on utiliser dès maintenant les balises `<audio>` et `<video>` ? Entre du Flash difficilement accessible et un standard HTML ouvert non implémenté partout, notre conseil sera d'utiliser ces balises HTML5 dès à présent, en fournissant une alternative (*fallback*), telle qu'un lien permettant de télécharger les fichiers médias et les fichiers de sous-titres, pour ceux qui n'ont pas accès à ces balises.

L'interactivité définie en HTML

Ce qui a changé sur le Web depuis l'introduction de HTML4, c'est l'interactivité, la propagation grand public de JavaScript, avec des bibliothèques de composants facilitant son utilisation, et le rapprochement de plus en plus intime entre pages web et applications avec l'avènement des applications web.

De nouvelles balises sémantiques, comme pour la structure, mais dédiées à la prise en compte des éléments d'interactivité, ont vu le jour. Nous pouvons citer, par exemple, `<progress>` qui permet de qualifier une barre de progression, `<meter>` qui donne lieu à une jauge, ou encore des attributs comme `contenteditable`, `draggable` ou `dropzone`... Tous ces éléments apportent du sens directement dans la page HTML, ce qui permettra aux technologies d'assistance d'exploiter ces informations.

Cependant, ces données n'ont aucune utilité sur une page statique. Une barre de progression indique, par exemple, le temps de traitement pour une action donnée. Il faut donc que l'information qu'elle contient (le temps restant) soit dynamiquement mise à jour. Or, cela passe le plus souvent par de l'Ajex, c'est-à-dire une mise à jour des données sans rechargement de la page via JavaScript.

Il était donc nécessaire, en plus d'un travail sur HTML5, de penser à rendre le code JavaScript accessible. C'est ce à quoi est destiné WAI-ARIA.

À NOTER WAI-ARIA valide uniquement à partir de HTML5

WAI-ARIA est une spécification prévue pour fonctionner à partir de HTML5. Ainsi, si vous codez votre page avec un type de document (**DOCTYPE**) antérieur, ne soyez pas étonné que le validateur du W3C considère les attributs ARIA comme autant d'erreurs.

Le mieux sera donc sans doute d'utiliser `<!DOCTYPE html>` en haut de vos pages, quitte à continuer à coder l'intérieur de façon « compatible avec l'accessibilité ».

WAI-ARIA, le Web riche accessible

En 2006, le groupe de travail WAI annonçait la création d'une suite de documents permettant aux développeurs de créer des contenus web dynamiques pouvant être utilisés par des personnes atteintes de handicap. WAI-ARIA était née.

À SAVOIR Spécification et prise en charge par les navigateurs

Vous pouvez retrouver la spécification WAI-ARIA sur le site du W3C :

► <http://ur1.ca/a1mvu>^a

Et pour connaître le niveau de prise en charge par les navigateurs, nous vous dirigeons vers le site déjà évoqué pour HTML5, dans sa section dédiée à WAI-ARIA :

► <http://caniuse.com/wai-aria>

a. <http://www.w3.org/TR/wai-aria/>

WAI-ARIA est une spécification du W3C conçue pour répondre à la question de l'accessibilité des contenus web riches, c'est-à-dire les applications web et les contenus dynamiquement mis à jour.

ARIA est l'abréviation de l'anglais *Accessible Rich Internet Applications suite*, ce qui signifie « suite accessible pour des applications Internet riches ». Cette spécification définit, via l'ajout de nouveaux attributs, un moyen de rendre les contenus et applications web plus accessibles aux personnes en situation de handicap. ARIA est prévue en particulier pour résoudre les questions d'accessibilité relatives aux contenus dynamiques et aux contrôles d'interface utilisateur avancés développés avec Ajax/JavaScript, HTML et autres technologies relatives à la conception de contenus riches.

VOCABULAIRE WAI-ARIA ou ARIA ?

WAI-ARIA est le nom officiel de la spécification. Pour éviter toute confusion, le W3C encourage l'utilisation du préfixe WAI associé à ARIA. Mais il n'est pas rare de trouver juste le mot ARIA. Si l'emploi de ce terme est fait dans le cadre d'un article sur l'accessibilité web, il s'agira sans doute de la même spécification. Pour faciliter la lecture, nous emploierons également la seule abréviation ARIA au sein des paragraphes qui suivent, à l'exception des en-têtes de section. Cette question a été abordée dans la Foire Aux Questions de la WAI dédiée à ARIA :
 ▶ <http://ur1.ca/a18yk^a>

a. <http://www.w3.org/WAI/aria/faq#justaria>

Comment ça marche ?

ARIA est extrêmement simple d'utilisation. Il s'agit d'ajouter de nouveaux attributs ARIA aux balises HTML. Il est également possible d'ajouter ces attributs sur d'autres spécifications, comme SVG, mais nous nous bornerons ici à parler de l'utilisation d'ARIA dans le cadre d'une page HTML.

Concrètement, voici comment ajouter de l'ARIA dans vos documents :

```
<li role="menuitemcheckbox"aria-checked="true">Classés par date</li>
```

Il s'agit donc d'associer à chaque balise un attribut ARIA et une valeur. Quand le contenu est amené à changer, il faut alors prévoir de modifier dynamiquement certaines valeurs, à l'aide de JavaScript, par exemple.

Dans l'exemple précédent, nous voyons que la balise d'élément de liste (`<li>`) a un rôle d'élément de menu que l'on peut cocher (`role="menuitemcheckbox"`). Et l'état de cette case est « cochée » (`aria-checked="true"`).

RESSOURCE En savoir plus sur l'utilisation d'ARIA

▶ <http://www.w3.org/TR/wai-aria/usage>

Il y a deux familles d'attributs ARIA :

- l'attribut `role`, qui précise le *rôle* de chaque élément, et vient donc apporter de la sémantique sur un élément qui n'en a peut-être pas ou qui mérite d'être précisé dans le contexte dans lequel il est employé ;
- les attributs commençant par le préfixe `aria-*` : ils permettent de définir des *propriétés* et des *états* pour les éléments sur lesquels ils sont ajoutés.

Rôles, propriétés, états : voyons tout cela en détail.

Les rôles, avec l'attribut `role`

Un rôle définit une fonction jouée par l'élément HTML dans un contexte donné.

ARIA définit trois types de valeurs pour l'attribut `role` :

- les valeurs venant décrire des zones de navigation (*landmarks*) ;
- les valeurs permettant de définir une structuration du document ;
- des valeurs identifiant des composants d'interfaces (*widgets*).

Il est ainsi possible de préciser le rôle d'un attribut HTML simplement en lui ajoutant l'attribut `role` et la valeur appropriée.

Les zones de navigation (*landmarks*)

Les zones de navigation servent à orienter l'utilisateur pour se représenter plus facilement la page s'il ne peut la voir. Dans l'exemple que nous avons pris plus haut concernant les nouvelles balises HTML5 de structuration de la page, voici ce que donnerait l'ajout des attributs ARIA :

```
<header role="banner">
  <hgroup>
    <h1>Titre de ma page</h1>
    <h2>Sous-titre de ma page</h2>
  </hgroup>
</header>
<div id="content" role="main">
  <article>
    <h3>Titre de l'article</h3>
    <p>Un premier paragraphe</p>
    <p>Un second paragraphe</p>
  </article>
  <article>
    <h3>Titre de l'article</h3>
    <p>Un premier paragraphe</p>
    <p>Un second paragraphe</p>
  </article>
</div><!--fin content-->
<aside role="complementary">
  <section>
    <h3>Édito du jour</h3>
    <p>Un premier paragraphe</p>
    <p>Un second paragraphe</p>
  </section>
  <section>
    <h3>Liens utiles</h3>
    <ul>
      <li><a href="/1">Lien 1</a></li>
      <li><a href="/2">Lien 2</a></li>
```

```
<li><a href="/3">Lien 3</a></li>
</ul>
</section>
</aside>
<footer role="contentinfo">
<p>Site réalisé par Môaa...</p>
</footer>
```

Vous noterez la simplicité d'utilisation... Ce dispositif pourra dispenser, à terme, l'ajout de liens d'accès rapide en haut de page pour se rendre directement sur les zones importantes de la page. Idéalement, les navigateurs eux-mêmes devraient exploiter ces informations pour fournir des raccourcis de navigation.

Étudions maintenant en détail les valeurs que peut prendre l'attribut `role` pour spécifier une zone de navigation.

Tableau 12-1 Liste des rôles ARIA pour les zones de navigation (landmark)

Valeur de l'attribut <code>role</code> pour définir des zones de navigation	Définition de la fonction du rôle
<code>application</code>	Définit une application web, à l'opposé d'un document web contenant des informations. Les technologies d'assistance agiront différemment s'il s'agit d'un document web ou d'une application (les raccourcis clavier ne seront pas les mêmes, les comportements non plus).
<code>banner</code>	Il s'agit d'un contenu spécifique au site entier, plutôt qu'à une page seulement, comme le titre du site, le logo, etc. Ces éléments, situés généralement en début de page web, peuvent être déclarés avec un <code>role="banner"</code> . Il ne peut y avoir qu'un seul élément dans la page marqué avec l'attribut <code>role="banner"</code> .
<code>complementary</code>	Il s'agit d'une zone de contenu liée à la zone principale, mais qui peut faire sens à elle toute seule, séparément. Il faut toutefois que cette zone apporte un complément d'information à la zone principale. Si ce n'est pas le cas, il est recommandé d'utiliser un rôle plus général.
<code>contentinfo</code>	Définit une zone relative au document parent et apportant des informations à son sujet. Typiquement, ce rôle peut être utilisé pour définir la zone (souvent en pied de page) contenant les conditions d'utilisation du site et les liens vers la politique de confidentialité. Il ne peut y avoir qu'un seul élément dans la page marqué avec l'attribut <code>role="contentinfo"</code> .

Tableau 12–1 Liste des rôles ARIA pour les zones de navigation (landmark) (suite)

Valeur de l'attribut role pour définir des zones de navigation	Définition de la fonction du rôle
form	La valeur <code>form</code> sert à définir une zone dans la page combinant différentes technologies qui, prises comme un tout, forment un formulaire. Une zone avec le rôle <code>form</code> peut être un mélange de contrôles de formulaires faits avec un langage serveur, des scripts et des liens. Ce n'est donc pas forcément relatif à la balise HTML <code><form></code> . Il est toutefois recommandé, quand c'est possible, d'utiliser les balises HTML natives plutôt qu'ARIA. Enfin, s'il s'agit spécifiquement d'un formulaire de recherche, il faut préférer le rôle plus précis <code>search</code> défini ci-après.
main	Définit la zone de contenu principal du document. Il ne peut y avoir qu'un seul élément dans la page marqué avec l'attribut <code>role="main"</code> .
navigation	Définit une collection d'éléments navigables (généralement des liens) pour naviguer dans le document ou les documents liés.
search	Il s'agit de préciser une zone contenant une collection d'éléments et d'objets qui, combinés et pris comme un tout, servent à faciliter la recherche.

RESSOURCE Liste des valeurs ARIA pour les zones de navigation (landmark role)

Vous pouvez consulter, en anglais, la liste des rôles ARIA et leurs valeurs pour indiquer les zones de navigation sur le site du W3C :

► <http://ur1.ca/a1956>^a

a. http://www.w3.org/TR/wai-aria/roles#landmark_roles

La structure de document

Certains rôles permettent de mieux identifier des éléments dans la page en les décrivant comme structurant le document. Par défaut, de nombreux éléments HTML possèdent un rôle implicite, natif. Il s'agit surtout des rôles de structuration du document. Par exemple, la balise HTML5 `<article>` est associée implicitement, c'est-à-dire sans même le préciser, à `role="article"`.

Les composants d'interfaces

Dans le cadre d'une application web, certaines parties de la page se comportent comme une application de bureau. C'est le cas, par exemple, pour une boîte de dialogue. Il s'agit alors de spécifier le rôle de chacun de ces composants pour que les

technologies d’assistance puissent savoir quel comportement adopter quant à l’utilisation des raccourcis clavier, par exemple.

Il y a deux types de rôles pour des composants d’interfaces :

- les composants d’interfaces simples ne nécessitant l’utilisation que d’un seul attribut `role` ;
- les composants d’interfaces complexes nécessitant la combinaison de plusieurs rôles pour être décrits.

Tableau 12–2 Valeurs de l’attribut `role` pour structurer un document HTML

Valeur de l’attribut <code>role</code>	Définition	Balise HTML pour laquelle la valeur de l’attribut <code>role</code> est associée par défaut (s’il y a lieu)
<code>article</code>	Section de la page qui constitue une composition indépendante du document, de la page ou du site. On peut donc la lire et la comprendre indépendamment du reste de la page.	<code><article></code>
<code>columnheader</code>	Cette valeur établit une relation entre elle et toutes les cellules correspondant à la colonne. Il faut s’assurer que les éléments ayant un <code>role="columnheader"</code> sont bien contenus dans un élément ayant un <code>role="row"</code> ou dépendent d’un élément avec un attribut <code>role="row"</code> .	<code><th scope="col"></code>
<code>definition</code>	Identifie la définition d’un terme ou d’un concept. Cet attribut ARIA ne permettant pas de préciser à quel terme cette définition est liée, il faut préciser le terme défini grâce aux attributs HTML <code>dfn</code> ou <code>dt</code> . Il est impératif d’identifier le terme défini par l’utilisation de l’attribut <code>aria-labelledby</code> sur chaque élément avec un rôle <code>definition</code> .	
<code>directory</code>	Sert à identifier une table des matières statique, que la liste des références soit faite sous forme de liens cliquables ou non. Par contre, dans le cas d’une table des matières dynamique, il faut utiliser le rôle <code>tree</code> plutôt que <code>directory</code> .	
<code>document</code>	Opposé au rôle <code>application</code> , il sert à indiquer les parties d’une page web qui servent à véhiculer une information et doivent être interprétées comme un document, ou des parties utilisées comme une application. Le rôle <code>document</code> est en général appliqué à l’ensemble de la page, sur la balise <code><body></code> par exemple.	

Tableau 12–2 Valeurs de l’attribut role pour structurer un document HTML (suite)

Valeur de l’attribut role	Définition	Balise HTML pour laquelle la valeur de l’attribut role est associée par défaut (s’il y a lieu)
group	Permet de grouper des éléments qui n’auraient pas leur place dans une table des matières, mais qui doivent être reliés. Ainsi, on ne met pas d’attribut <code>role="group"</code> sur des balises de titres, qui sont suffisamment explicites et peuvent figurer dans une table des matières. Par contre, on peut utiliser le rôle <code>group</code> pour s’assurer qu’une image et sa légende seront liées en cas de non prise en compte des balises <code><figure></code> et <code><figcaption></code> .	
heading	Définit un en-tête pour une section de la page.	<code><h1></code> , <code><h2></code> , <code><h3></code> , <code><h4></code> , <code><h5></code> , <code><h6></code>
img	Définit un bloc contenant une collection d’éléments qui forment une image. Un rôle <code>img</code> peut être attribué à un bloc contenant des légendes et un texte descriptif, ainsi que de multiples fichiers images qui, vus ensemble, donnent l’impression de ne former qu’une seule image. Un bloc avec l’attribut <code>role="img"</code> représente un élément graphique unique au sein d’un document, qu’il s’agisse ou non d’une collection d’objets dessinés.	<code></code>
list	Groupe d’éléments de liste, non interactifs. Les éléments enfants contenu dans la balise recevant le rôle <code>list</code> ont forcément le rôle <code>listitem</code> .	<code></code> , <code></code>
listitem	Élément de liste seul, contenu dans une liste ou une table des matières. Un élément ayant pour attribut <code>role="listitem"</code> doit forcément être contenu dans un élément ayant pour attribut <code>role="list"</code> .	<code></code>
note	Définit une section dont le contenu constitue une parenthèse ou un contenu auxiliaire par rapport au contenu principal. C’est le cas, par exemple, pour une note de bas de page.	

Tableau 12–2 Valeurs de l’attribut role pour structurer un document HTML (suite)

Valeur de l’attribut role	Définition	Balise HTML pour laquelle la valeur de l’attribut role est associée par défaut (s’il y a lieu)
presentation	Cette valeur est à utiliser avec précaution, car elle a pour objet de supprimer le rôle natif de l’élément, afin que ce rôle ne soit pas restitué par une technologie d’assistance. Ainsi : <pre><table role="presentation"> <tr> <td>Exemple</td> </tr> </table></pre> sera restitué de la façon suivante : <pre> Exemple </pre> Il s’agit donc de désigner un élément dont le sens du rôle natif ne doit pas être retransmis aux API d’accessibilité. L’utilisation visée concerne un élément qui modifie l’apparence de la page, mais dans le cas où cet élément ne se sert pas de toutes les caractéristiques de la balise (fonctions, interactions, structure), ou un élément qui assure une alternative accessible dans les vieux navigateurs non compatibles avec WAI-ARIA. C’est le cas, par exemple, pour un élément dont le contenu est uniquement décoratif (une image d’espace-ment, une illustration décorative ou un élément pour permettre un saut de ligne – <code>clear</code>). Autre exemple : une image contenue dans un bloc avec le rôle <code>img</code> et au sein duquel une alternative textuelle complète est disponible et marquée avec les attributs <code>aria-labelledby</code> et (si nécessaire) <code>aria-describedby</code>. Enfin, dernier exemple, un tableau de mise en page utilisé à seule fin de présentation.	

Tableau 12–2 Valeurs de l’attribut role pour structurer un document HTML (suite)

Valeur de l’attribut role	Définition	Balise HTML pour laquelle la valeur de l’attribut role est associée par défaut (s’il y a lieu)
region	Permet de définir une région dans la page qui mérite d’être décrite comme une zone de contenu à part entière. Cela ressemble aux rôles définissant les zones de navigation (<i>landmark roles</i>), mais dont les valeurs associées ne permettent pas de décrire la région en question. Cette zone a vocation à être restituée dans une table des matières telle que peuvent en fabriquer certaines technologies d’assistance. Il faut s’assurer que pour chaque région, un en-tête est référencé avec l’attribut <code>aria-labelledby</code> .	<frame>
row	Définit une rangée au sein de laquelle sont contenues des cellules (<code>gridcell</code>) afin d’organiser une grille (<code>grid</code>), comme dans un tableau.	<tr>
rowheader	Définit une cellule contenant une information d’en-tête pour une ligne dans une grille.	<th scope="row">
separator	Définit une séparation visuelle qui distingue des sections de contenu ou des groupes d’éléments dans un menu.	<hr />
toolbar	Désigne une barre d’outils, c’est-à-dire une collection de boutons fonctionnels couramment utilisés et représentés sous une forme visuelle compacte. L’élément avec le rôle <code>toolbar</code> est souvent un sous-ensemble de fonctions que l’on retrouve dans un élément avec l’attribut <code>role="menubar"</code> , dans le but de réduire l’effort dans l’utilisation de ces fonctions. Il est nécessaire de fournir une propriété <code>aria-label</code> sur chaque barre d’outils quand l’application en contient plus d’une.	

ATTENTION Ne jamais aller contre le rôle natif d’une balise HTML

Nous venons de voir que certaines balises HTML avaient un rôle natif. Ces rôles peuvent être surchargés pour préciser le rôle de l’élément. Ainsi, une balise `<article>`, qui a un rôle ARIA natif avec la valeur `article`, peut se voir attribuer une autre valeur plus précise dans le contexte d’utilisation de l’élément, comme un `role="main"`, par exemple.
Cependant, *il est interdit de modifier le rôle natif de l’élément* pour aller à l’encontre du comportement sémantique prévu en HTML. Cela constitue une erreur de codage et un détournement de l’usage des balises.

L'utilisation d'ARIA sert avant tout à apporter du sens sur des éléments qui n'en ont pas (`div`, `span`), à préciser un rôle sur une balise dont la fonction peut être affinée avec ARIA, ou à pallier certaines insuffisances des navigateurs ou technologies d'assistance, qui prennent parfois mieux en compte les rôles ARIA que les nouvelles balises HTML5.

Sans entrer dans le détail de la définition de chaque valeur, voici la liste des rôles décrivant des composants d'interfaces utilisateurs isolés ou agissant comme des composants d'interfaces composites au sein d'un ensemble plus large :

- `alert`
- `alertdialog`
- `button`
- `checkbox`
- `dialog`
- `gridcell`
- `link`
- `log`
- `marquee`
- `menuitem`
- `menuitemcheckbox`
- `menuitemradio`
- `option`
- `progressbar`
- `radio`
- `scrollbar`
- `slider`
- `spinbutton`
- `status`
- `tab`
- `tabpanel`
- `textbox`
- `timer`
- `tooltip`
- `treeitem`

Les rôles suivants concernent des composants d'interfaces plus complexes. Ils agissent en général comme un bloc contenant d'autres composants :

- `combobox`
- `grid`

- `listbox`
- `menu`
- `menubar`
- `radiogroup`
- `tablist`
- `tree`
- `treegrid`

RESSOURCE Les rôles ARIA pour composants d'interfaces

Pour en savoir plus sur l'utilisation de chaque attribut, vous pouvez consulter la documentation officielle (en anglais) sur le site du W3C :

► <http://ur1.ca/a1d6y^a>

a. http://www.w3.org/TR/wai-aria/roles#widget_roles

Propriétés et états

Alors que les attributs `role` permettent d'indiquer quelle est la *nature* des balises associées, une autre famille d'attributs ARIA permet de préciser des propriétés spécifiques pour cette balise, ainsi que son état à un instant *t*.

La différence entre propriétés et états est subtile, selon la spécification. Ces deux types d'attributs permettent de fournir des informations spécifiques à propos d'un objet, et tous deux permettent de préciser la définition apportée par la nature des rôles.

Les deux types d'attributs sont traités avec le préfixe `aria-*`. Mais il y a une différence d'appellation, malgré tout, en raison de leur aspect stable (pour les propriétés) ou changeant (pour les états). Ainsi, une propriété telle que `aria-labelledby` indique que la balise est liée à une légende. Cette propriété n'a pas vocation à changer fréquemment. En revanche, un état tel que `aria-checked` est prévu pour changer en fonction de l'action de l'utilisateur.

La fréquence de changement n'est cependant pas une règle permettant de distinguer propriétés et états. Par exemple, certaines propriétés telles que `aria-activedescendant`, `aria-valuenow` et `aria-valuetext` ont vocation à changer souvent.

Cette distinction n'a pas vraiment de conséquence sur l'utilisation des attributs, que nous allons voir maintenant en détail.

Valeurs possibles pour les propriétés et états ARIA

Les attributs indiquant un état (`state`) ou une propriété (`property`) peuvent avoir les valeurs suivantes.

Tableau 12–3 Valeurs des états (state) et propriétés (property)

Valeurs possibles	Description
true/false	Valeur représentant soit vrai, soit faux, avec une valeur <code>false</code> par défaut.
tristate	Valeur représentant vrai ou faux, avec une valeur intermédiaire mixte. Par défaut, la valeur est <code>false</code> , à moins qu'il ait été spécifié autre chose.
true/false/undefined	Valeur représentant vrai ou faux, avec une valeur par défaut <code>undefined</code> indiquant que l'état ou la propriété n'est pas pertinent.
Un identifiant <code>id</code>	Une référence à l'attribut <code>id</code> d'un autre élément dans le même document.
Une liste d'identifiants <code>id</code>	Une liste d'un ou plusieurs attributs <code>id</code> .
Un nombre entier	Une valeur numérique non fractionnée.
Un nombre	Une valeur quelconque d'un nombre réel.
Une chaîne (<code>string</code>)	Une valeur libre.
Un segment d'un langage de programmation (<code>token</code>)	L'une des valeurs autorisées parmi un nombre limité.
Une liste de segments d'un langage de programmation	Une liste d'un ou plusieurs éléments (<code>token</code>).

Les propriétés et états sont déclinés en quatre catégories concernant :

- les **régions dynamiques** (*live regions*) : ce sont les parties d'un contenu ou d'une application web qui se mettent à jour sans que la page soit rechargée ;
- le **glisser-déposer** (*drag&drop*) : il s'agit de rendre accessibles ces zones où l'on peut « saisir » un objet et le déplacer vers un autre endroit de la page, ce qui se fait en principe avec la souris ;
- les **misés en relation** : il s'agit de préciser la relation entre deux éléments, comme une image et sa légende ;
- les **composants d'interfaces** (*widgets*) : ils sont alors liés à un rôle définissant les composants d'interfaces.

Les régions dynamiques (live regions)

L'un des grands défis du Web riche pour la mise en accessibilité, c'est que même si les technologies d'assistance récentes peuvent maintenant détecter qu'un élément dans la page a été modifié alors que la page elle-même n'a pas été rechargée (typiquement avec Ajax), il ne leur est pas possible de savoir où et de quelle façon ce changement a affecté le contenu.

Un exemple simple et très fréquent : le widget fourni par Twitter permettant d’afficher le flux des derniers tweets publiés. Il permet l’affichage en temps réel des nouveaux tweets publiés, sans que la page web sur laquelle figure le widget soit elle-même rechargée.

L’objet des attributs ARIA relatifs aux régions dynamiques est d’indiquer que *des changements de contenu peuvent se produire sans que l’élément en question ait le focus*, et d’indiquer aux technologies d’assistance *la façon dont elles doivent rendre compte de ces mises à jour*.

Les différents attributs présentés dans le tableau ci-après permettent de préciser aux logiciels de lecture d’écran la façon dont ils doivent, ou non, restituer l’information.

Tableau 12–4 Liste des attributs ARIA relatifs aux régions dynamiques

Attribut ARIA	Définition	Valeurs possibles
<code>aria-atomic</code>	Indique si les technologies d’assistance doivent présenter tout, ou seulement une partie, de la région modifiée en fonction des notifications définies par l’attribut <code>aria-relevant</code> . Quand le contenu d’une région dynamique change, les agents utilisateurs doivent examiner l’élément qui a changé et remonter les ancêtres de l’élément pour trouver le premier élément avec l’attribut <code>aria-atomic</code> , puis appliquer le comportement pour les cas qui suivent. Quand <code>aria-atomic</code> a pour valeur <code>true</code> , les technologies d’assistance peuvent choisir de combiner plusieurs changements ayant lieu dans la zone et de les présenter en une seule fois, comme un tout.	<code>true</code> : les technologies d’assistance présenteront la région comme un tout. <code>false</code> (par défaut) : seul le changement dans la région sera transmis par les technologies d’assistance, sans son contexte.
<code>aria-busy</code> (<code>state</code>)	Indique si la mise à jour de la région est terminée ou toujours en cours. Si la valeur est <code>true</code> , cela signifie que la zone n’a pas fini de charger les mises à jour.	<code>true</code> : la région dynamique est toujours en attente de nouvelles mises à jour. <code>false</code> (par défaut) : il n’y a pas de nouvelles mises à jour attendues pour la zone.

Tableau 12–4 Liste des attributs ARIA relatifs aux régions dynamiques (suite)

Attribut ARIA	Définition	Valeurs possibles
<code>aria-live</code>	Indique que la région est une région qui sera mise à jour. Selon la valeur de l'attribut, l'indication porte aussi sur la façon dont l'information mise à jour doit être restituée.	off (par défaut) : les mises à jour de la zone ne seront pas retransmises à l'utilisateur, à moins que son focus soit placé sur la zone. polite : la technologie d'assistance annoncera la mise à jour dans la zone dès que l'opportunité se présentera sans que cela gêne l'utilisateur ; par exemple, à la fin d'une phrase ou quand l'utilisateur arrête de saisir un texte. assertive : cette information a la priorité la plus élevée et les technologies d'assistance notifieront la mise à jour de la zone dès qu'elle se produit, même si l'utilisateur est en train de lire un texte. Cette propriété peut perturber l'utilisateur ; elle ne doit donc être utilisée que si l'interruption s'impose, comme pour une alerte de sécurité, par exemple.
<code>aria-relevant</code>	Indique quelles modifications il convient de transmettre à l'utilisateur via sa technologie d'assistance. Il s'agit de modifications ayant du sens, contrairement aux modifications ne concernant que la présentation. Par exemple, si un bout de conversation disparaît de l'écran au cours d'un tchat par Internet (IRC), pour faire de la place, ce n'est pas une information ayant du sens. Cela ne vise qu'à mieux répartir l'information visuellement. En revanche, si dans la liste des gens connectés au salon IRC, un nom disparaît, cela signifie que la personne est déconnectée. Il est important que cette information, qui a une signification, soit retransmise.	additions : les nœuds de l'élément sont additionnés au DOM au sein de la région dynamique, ce qui permet de remonter le contexte de l'élément actif. removals : le texte ou les nœuds de l'élément à l'intérieur de la région sont enlevés du DOM. text : le texte est ajouté à n'importe quel nœud descendant du DOM dans la région dynamique. all : équivalent à une combinaison de toutes les valeurs (" additions removals text "). additions text (par défaut) : équivalent à l'addition des valeurs " additions text ".

RESSOURCE Les propriétés et états ARIA pour régions dynamiques

Pour en savoir plus sur l'utilisation de chaque attribut, vous pouvez consulter la documentation officielle (en anglais) sur le site du W3C :

► <http://ur1.ca/a1d63>^a

a. http://www.w3.org/TR/wai-aria/states_and_properties#attrs_liveregions

Le glisser-déposer (drag&drop)

Cette section concerne les interfaces proposant aux utilisateurs de glisser-déposer des éléments. Il s’agit de saisir un objet (en principe à la souris) et de le déposer dans une zone différente. L’information quant à l’endroit où sera déposé l’élément (la cible du dépôt) sera rendue visuellement, mais également fournie aux technologies d’assistance à travers une modalité alternative.

Tableau 12–5 Liste des attributs ARIA relatifs au glisser-déposer

Attribut ARIA	Définition	Valeurs possibles
<code>aria-dropeffect</code>	Indique quelles sont les actions pouvant être réalisées quand l’objet saisi est relâché dans la zone cible de dépôt. Plusieurs valeurs peuvent être utilisées simultanément pour un même attribut. En plus de l’attribut <code>aria-dropeffect</code> , une indication visuelle des cibles de dépôt doit être prévue.	<code>copy</code> : une copie de l’objet source sera déposée dans la cible. <code>move</code> : l’objet source sera enlevé de sa position actuelle et déposé dans la cible. <code>link</code> : une référence ou un raccourci vers l’objet déposé sera créé dans l’objet cible. <code>execute</code> : une fonction implémentée par la cible du dépôt est exécutée, utilisant la source de saisie (telle qu’un <code>input</code>). <code>popup</code> : une pop-up de menus ou une boîte de dialogue permet à l’utilisateur de choisir une des opérations de saisie (<code>copy</code> , <code>move</code> , <code>link</code> , <code>execute</code>) et une des autres fonctionnalités de saisie, comme « annuler ». <code>none</code> (par défaut) : aucune opération ne peut être accomplie. En pratique, cela annule l’opération de saisie si une tentative est faite pour déposer l’objet.
<code>aria-grabbed (state)</code>	Indique si l’élément est dans l’état « saisi » (valeur <code>true</code>) ou non (valeur <code>false</code>) pour effectuer une opération de glisser-déposer.	<code>true</code> : indique que l’élément a été saisi pour être déposé. <code>false</code> : indique que l’élément peut être saisi. <code>undefined</code> (par défaut) : indique que l’élément ne peut pas être saisi.

RESSOURCES Les propriétés et états ARIA pour le glisser-déposer

Pour en savoir plus sur l’utilisation de chaque attribut, vous pouvez consulter la documentation officielle (en anglais) sur le site du W3C :

► <http://ur1.ca/a1d68>^a

Pour plus d’informations sur l’utilisation du glisser-déposer avec ARIA, voir la traduction française par Cédric Trévisan d’un article de Gez Lemon :

► <http://ur1.ca/a1mwd>^b

a. http://www.w3.org/TR/wai-aria/states_and_properties#attrs_dragdrop
b. http://www.paciellogroup.com/blog/misc/ddfrench/WAI-ARIA_DragAndDrop_French.html

Les mises en relation

Cette section liste les attributs permettant d’indiquer les relations ou associations entre des éléments qui ne peuvent pas être établies facilement à partir de la structure du document.

Tableau 12–6 Liste des attributs ARIA relatifs aux mises en relation dans des documents HTML

Attribut ARIA	Définition	Valeurs possibles
<code>aria-controls</code>	Identifie l’élément (ou les éléments) dont le contenu ou la présence est contrôlé par l’élément ayant l’attribut <code>aria-controls</code> . Par exemple, un onglet contrôle l’affichage du panneau d’onglets.	Liste d’identifiants <code>id</code>
<code>aria-describedby</code>	Identifie l’élément ou les éléments qui décrivent l’objet. Cela ressemble beaucoup au fait de mettre une légende sur un objet avec <code>aria-labelledby</code> . Une légende fournit à l’utilisateur la nature de ce que l’objet fait, alors qu’une description a pour but de fournir <i>des détails supplémentaires</i> dont certains utilisateurs pourraient avoir besoin.	Liste d’identifiants <code>id</code>
<code>aria-flowto</code>	Identifie le prochain élément à lire selon un ordre de lecture alternatif à celui proposé par défaut dans la page. Plusieurs identifiants, et donc plusieurs chemins, peuvent être proposés. Les technologies d’assistance doivent laisser le choix à l’utilisateur d’utiliser l’un de ces chemins ou de poursuivre selon l’ordre de lecture naturel.	Liste d’identifiants <code>id</code>
<code>aria-labelledby</code>	Identifie l’élément ou les éléments qui donnent une légende à l’élément porteur de l’attribut <code>aria-labelledby</code> . Cela permet de donner à l’objet un nom que l’utilisateur peut identifier.	Liste d’identifiants <code>id</code>
<code>aria-owns</code>	Identifie un élément ou des éléments afin de définir une relation parent/enfant visuelle, fonctionnelle ou contextuelle, entre des éléments du DOM pour lesquels la hiérarchie du DOM ne peut être utilisée pour représenter la relation. La raison pour ajouter <code>aria-owns</code> est d’exposer la relation parent/enfant aux technologies d’assistance qui, sans cela, ne pourraient pas la déduire.	Liste d’identifiants <code>id</code>

Tableau 12–6 Liste des attributs ARIA relatifs aux mises en relation dans des documents HTML (suite)

Attribut ARIA	Définition	Valeurs possibles
<code>aria-posinset</code>	Définit le nombre ou la position d'un élément au sein d'un ensemble d'éléments de liste ou d'éléments dans une arborescence. Si tous les éléments sont présents dans la structure du document, il n'est pas nécessaire d'utiliser cet attribut, l'agent utilisateur pouvant calculer automatiquement la taille et la position de chaque élément. Par contre, si seulement une portion de l'ensemble est présent dans la structure du document à un moment donné, cette propriété est nécessaire pour fournir une indication explicite de la position de l'élément.	Nombre entier
<code>aria-setsize</code>	Comme l'attribut <code>aria-posinset</code> , il définit la taille d'un élément au sein d'un ensemble d'éléments de liste ou d'éléments dans une arborescence. Il décrit donc le nombre d'éléments total. Cette propriété est utilisée sur les membres d'un ensemble, pas sur le bloc lui-même contenant les éléments. Pour orienter l'utilisateur en disant que l'élément est « l'élément X sur Y », les technologies d'assistance utiliseraient X comme équivalent à l'attribut <code>aria-posinset</code> et Y comme étant égal à <code>aria-setsize</code> .	Nombre entier

L'attribut `aria-activedescendant` n'a pas été traité, car il ne concerne pas les documents HTML, mais uniquement les applications.

RESSOURCE Les propriétés et états ARIA pour les mises en relation d'éléments

Pour en savoir plus sur l'utilisation de chaque attribut, vous pouvez consulter la documentation officielle (en anglais) sur le site du W3C :

► <http://ur1.ca/a1d6c>^a

a. http://www.w3.org/TR/wai-aria/states_and_properties#attrs_relationships

Les composants d'interfaces (widgets)

Comme précédemment, nous n'entrerons pas dans le détail concernant les composants d'interfaces. Mais voici la liste des attributs permettant de spécifier la propriété et l'état d'un widget. Quand il s'agit d'un état, il est indiqué `state` entre parenthèses à côté de l'attribut concerné.

- `aria-autocomplete`
- `aria-checked` (state)
- `aria-disabled` (state)
- `aria-expanded` (state)
- `aria-haspopup`
- `aria-hidden` (state)
- `aria-invalid` (state)
- `aria-label`
- `aria-level`
- `aria-multiline`
- `aria-multiselectable`
- `aria-orientation`
- `aria-pressed` (state)
- `aria-readonly`
- `aria-required`
- `aria-selected` (state)
- `aria-sort`
- `aria-valuemax`
- `aria-valuemin`
- `aria-valuenow`
- `aria-valuetext`

RESSOURCE Les propriétés et états ARIA pour composants d'interfaces

Pour en savoir plus sur l'utilisation de chaque attribut, vous pouvez consulter la documentation officielle (en anglais) sur le site du W3C :

► <http://ur1.ca/a1d5a>^a

a. http://www.w3.org/TR/wai-aria/states_and_properties#attrs_widgets

WAI-ARIA, « la » solution ?

Nous avons vu que la spécification WAI-ARIA était finalement une couche supplémentaire que l'on ajoute aux balises déjà présentes pour *donner du sens et gérer un comportement*.

Il y a trois cas d'utilisation d'ARIA :

- pour ajouter du sens à une balise qui n'en a pas (`<div>`, `<span>`), par exemple `<div role="slider">` ;

- pour préciser le sens d'une balise en lui apportant un niveau de détail supplémentaire (`<article role="main">`) ;
- pour permettre une meilleure compatibilité des navigateurs et des technologies d'assistance avec HTML5, ceux-ci implémentant parfois mieux ARIA que certaines nouvelles balises HTML5 (`<nav role="navigation">`). Évidemment, il est inutile d'ajouter des attributs ARIA redondants pour les balises HTML4 déjà connues des navigateurs et des technologies d'assistance. Écrire `<a role="link">` n'a aucune utilité.

ASTUCE Insérer automatiquement des rôles ARIA

Voici un script pour insérer automatiquement des rôles ARIA sur certaines des nouvelles balises HTML5 :

► <http://ur1.ca/a1gck^a>

a. <https://github.com/yatil/accessifyhtml5.js/>

Or, la simplicité d'utilisation des attributs ARIA, qui s'ajoutent comme n'importe quel attribut sur une balise HTML, cache une double difficulté.

- Il faut, d'une part, bien connaître ces attributs et les valeurs qu'ils peuvent prendre pour les utiliser à bon escient.
- Il convient, d'autre part, de faire varier de façon appropriée, grâce à JavaScript, les valeurs de ces attributs, et notamment des états qui ont vocation à changer selon l'action de l'utilisateur.

Pour améliorer l'accessibilité de certaines infrastructures web lourdes, qui mériteraient une refonte complète du code HTML, la tentation est grande d'injecter de l'ARIA pour modifier le comportement natif des éléments. ARIA est alors utilisée comme une « rustine » pour rendre accessibles des contenus qui ne le sont pas.

Nous attirons votre attention sur le fait que cette pratique, qui peut donner de bons résultats immédiats à moindre coût, est loin d'être idéale. Elle témoigne en effet d'une prise en compte de l'accessibilité « après coup », une fois que le gabarit HTML a déjà été créé. Cela signifie sans doute que de nombreuses questions relatives à la structure des pages, aux contenus, à la navigation ou à la formation des contributeurs n'auront pas été traitées. L'injection d'attributs ARIA ne pourra pas corriger toutes ces questions comme par magie. La mise en accessibilité ne pourra être que technique et parcellaire.

Par ailleurs, l'ajout de rôles ARIA ne supprime pas le comportement natif des éléments. Prenons par exemple `<button role="heading">titre</button>`. Ce bouton transformé en titre sera bien restitué par le lecteur d'écran comme un titre, mais il

restera cliquable et on pourra l'atteindre au clavier, tout en étant présenté visuellement par le navigateur comme un bouton.

ATTENTION Des composants JavaScript accessibles au clavier

ARIA ne gère pas le comportement au clavier. Tous les composants d'interfaces réalisés en JavaScript devront être prévus pour être accessibles au clavier. Une gestion du focus via l'ajout en JavaScript d'attributs `tabindex` avec une valeur changeant selon les besoins est toujours nécessaire.

Pour information : l'attribut `tabindex="0"` donne le focus à l'élément, tandis que l'attribut `tabindex="-1"` supprime le focus de l'élément. La valeur de `tabindex` sera gérée en JavaScript.

N'attendez donc pas qu'ARIA soit la solution miracle à tous vos problèmes d'accessibilité. Rien ne remplacera un code HTML structuré avec les bonnes balises sémantiques.

En revanche, employée à bon escient, ARIA est *une spécification incontournable pour faire du Web riche accessible*. Alors qu'il était impensable d'offrir une fonctionnalité de glisser-déposer à tous vos utilisateurs – la seule solution pour l'accessibilité étant de faire une alternative, ce qui n'est pas toujours simple –, ARIA rend possible le glisser-déposer accessible pour tous. Et ce n'est qu'un exemple !

L'intérêt majeur de cette spécification est qu'elle peut être utilisée dès à présent. Car quel que soit la prise en charge de cette spécification par les navigateurs ou les technologies d'assistance, le risque pour vos utilisateurs est nul. Dans le pire des cas, ils ne bénéficieront pas de ce qu'apporte ARIA. C'est ce que dit d'ailleurs la WAI : « Dans les navigateurs qui n'implémentent pas WAI-ARIA, les contenus web qui ajoutent des attributs WAI-ARIA continueront simplement à fonctionner comme ils le font actuellement dans ces navigateurs. »² Et au mieux, des informations supplémentaires seront apportées. Alors, n'hésitez pas !

RESSOURCES Articles sur WAI-ARIA

Démonstration de composants jQuery accessibles avec ARIA (en anglais) :

► <http://ur1.ca/a1mx9^a>

Article de synthèse illustré d'exemples par Aurélien Levy sur letrainde13h37.fr :

► <http://ur1.ca/9ptxl^b>

Annexe sur ARIA par Rodople Rimelé, en libre téléchargement sur le site de l'éditeur :

► <http://ur1.ca/a1mxr^c> (PDF, 2,1 Mo)

a. http://hanshillen.github.com/jqtest/#goto_slider

b. <http://letrainde13h37.fr/8/aria-il-serait-temps-de-sy-mettre/>

c. http://www.eyrolles.com/Chapitres/9782212129823/Annexe-C_Rimele.pdf

2. <http://www.w3.org/WAI/aria/faq#browsers>

Et maintenant, que faire ?

Nous n'avons cessé de vous mettre en garde sur le fait de s'assurer de la compatibilité de HTML5 avec l'accessibilité, tout en vous vantant ses avancées prometteuses. Alors, que faire ?

La WAI a rédigé les WCAG 2 pour qu'elles ne reposent pas sur une technologie en particulier. HTML5 n'est donc pas interdit. D'un autre côté, parmi les techniques permettant de valider les critères de succès, aucune ne porte encore sur HTML5. Or, le processus de validation d'une nouvelle technique implique de nombreux tests avec différents agents utilisateurs (navigateurs et technologies d'assistance) pour s'assurer que cela fonctionne correctement – et largement. L'utilisateur est donc au centre de ce processus. Et à ce jour, aucune technique HTML5 n'a encore été validée, car aucune n'est suffisamment implémentée. Cela devrait peut-être changer en 2013...

Que faire alors en attendant que HTML5 soit considéré comme « compatible avec l'accessibilité » ? Le mieux que nous puissions vous conseiller, c'est de coder intelligemment, en prévoyant une alternative accessible dans les cas où les utilisateurs n'ont pas accès à HTML5.

Les règles d'accessibilité elles-mêmes restent pertinentes dans leur principe, il ne vous sera donc pas difficile de vérifier l'accessibilité de votre code avec AccessiWeb 2.2, même si le référentiel est basé sur HTML4.

Il serait dommage, de notre point de vue, de ne pas profiter des avancées de HTML5. Anticipez donc en utilisant le bon Doctype, continuez toutefois à coder vos titres selon une logique linéaire HTML4, balisez vos documents en profitant de l'apport sémantique des balises HTML5 couplées à de l'ARIA... et tout ira bien !

À NOTER Le cas des liens en HTML5

Il est nécessaire, pour le moment, de respecter la sémantique `inline/block` de HTML4. C'est surtout vrai pour la balise de lien `<a>`. HTML5 permet de faire des liens sur à peu près tout élément. La balise `<a>`, qui en HTML4 était une balise de type `inline`, peut être utilisée comme une balise de type `block` en HTML5. Or, à l'heure où nous écrivons ces lignes, ce comportement n'est pas géré par les technologies d'assistance. Par exemple, un titre (balise `<h>`) encadré par des balises de lien `<a>` traitées comme des balises de type `block` n'est plus restitué dans la hiérarchie de titres de la page.

Par ailleurs, la restitution d'un élément `<a>` pouvant contenir ce qu'on veut risque de poser problème pour interpréter l'intitulé du lien. Une solution sera sans doute de faire appel à l'attribut `aria-labelledby`, qui pourrait permettre d'indiquer quel est le passage de texte à restituer comme intitulé. Cette nouveauté HTML5 pose dans tous les cas de nouveaux défis pour l'accessibilité. Il sera intéressant de voir comment elle sera traitée par les agents utilisateurs. Pour le moment, elle n'est pas compatible avec l'accessibilité.

Pour respecter le référentiel AccessiWeb 2.2, il n'y a d'autre choix que de coder en HTML4 pour le moment. Car utiliser le type de document HTML5 implique que certains éléments (balises et attributs) sont obsolètes. Or, si vous n'utilisez pas d'attribut `summary` sur un tableau (élément obsolète en HTML5), vous invalidez AccessiWeb (le critère 5.1). Et si vous utilisez cet élément sur un `DOCTYPE` HTML5, vous invalidez le code et donc le critère AccessiWeb 8.2 (le test 8.2.2 précisément).

INFO Labelliser un site en HTML5 avec AccessiWeb 2.2

Vous connaissez le référentiel AccessiWeb, mais il existe aussi le label AccessiWeb qui est délivré sur la base d'une évaluation de la conformité d'un échantillon de pages web au regard des critères du référentiel AccessiWeb.

Or, AccessiWeb 2.2 est incompatible avec HTML5. Cela signifie-t-il qu'il est impossible de demander un label AccessiWeb si votre site est en HTML5 ? À vrai dire, non. BrailleNet est en train de travailler sur la question et effectue déjà des labellisations sur des sites en HTML5 à titre expérimental. Une procédure spécifique est à l'étude en attendant de pouvoir disposer d'une procédure ou d'un référentiel dédié à HTML5.

Pour en savoir plus sur la procédure de labellisation, vous pouvez consulter le site AccessiWeb :

► <http://ur1.ca/a5d60a>

a. <http://www.accessiweb.org/index.php/procedure-de-labellisation.html>

En résumé, rien ne vous empêche de faire de l'accessibilité avec une utilisation raisonnée de HTML5, alors n'hésitez pas !

POUR ALLER PLUS LOIN L'accessibilité au delà des contenus web

Pour clore cette deuxième partie du livre consacrée à l'application pratique de l'accessibilité des contenus web, nous ne pouvons faire l'impasse sur deux autres sujets connexes : l'accessibilité des outils d'édition, comme les logiciels de gestion de contenus (CMS) et l'accessibilité des documents en téléchargement, notamment les documents bureautiques.

Ce sujet aurait dû faire l'objet d'un chapitre entier, voire même de deux chapitres distincts. Le temps ayant manqué avant publication, nous ne l'aborderons ici que succinctement, en vous fournissant les liens pour vous documenter.

L'accessibilité des outils d'édition

Un référentiel spécifique a été conçu par la WAI à destination des concepteurs d'outils d'édition (*authoring tools*, en anglais). Il s'agit du référentiel ATAG (*Authoring Tools Accessibility Guidelines*), soit les règles d'accessibilité pour les outils d'édition.

Sur le Web, on pensera bien sûr aux CMS. Mais ces règles s'appliquent également aux concepteurs de traitements de texte, par exemple.

Non seulement l'outil doit être accessible et permettre la production de contenus accessibles, mais il doit également accompagner le contributeur pour qu'il puisse créer des contenus accessibles.

Alors que ATAG 1 repose sur les WCAG 1, ATAG 2 sera compatible avec les WCAG 2. Pourquoi « sera » ? Parce que la version 2 d'ATAG est encore en cours de finalisation. La version définitive est annoncée pour l'année 2013. Nous vous encourageons tout de même à vous reporter à la version 2 d'ATAG, même non finalisée, disponible en ligne :

► <http://www.w3.org/TR/ATAG20/>

L'accessibilité des documents en téléchargement

Quand un document est en téléchargement sur une page web, soit l'information est également présente dans la page et accessible, auquel cas le document bureautique n'est qu'une alternative non accessible à la page web, soit l'information est seulement contenue dans le document en téléchargement, et alors, ce document doit être accessible.

Vous trouverez, en annexe du référentiel AccessiWeb 2.2 déployé, un document à télécharger reprenant la liste des règles applicables pour vérifier l'accessibilité d'un document bureautique :

► <http://ur1.ca/9yks3^a>

Il est toujours mieux de penser ses documents bureautiques accessibles dès la conception, dans votre logiciel de traitement de texte par exemple, avant de les transformer en PDF. Mais si vous êtes amenés à travailler sur des documents PDF déjà faits, et que vous souhaitez les baliser pour les rendre accessibles, vous serez contraints d'utiliser le logiciel Adobe Acrobat Pro. Une notice très complète est à disposition sous licence libre d'utilisation dans un contexte non commercial. Il s'agit de la notice AcceDe PDF éditée par la société Atalan et disponible sur le site dédié :

► <http://ur1.ca/a8ck9^b>

a. http://www.accessiweb.org/index.php/accessiweb_2.2_liste_deployee.html#annexe

b. <http://www.pdf-accessible.com/fr/pdf-et-accessibilite/>

TROISIÈME PARTIE

L'accessibilité du Web, au-delà des normes

Aïe, aïe, aïe... Vous venez de lire déjà plus de trois cents pages sur l'accessibilité web et le mal de tête vous guette ? Pas de panique, cette partie est là pour vous faire prendre un peu de recul sur toutes ces règles et voir par où commencer !

La démarche accessibilité

跋鰲千里

« Même une tortue boiteuse peut parcourir mille lieues. »

Proverbe chinois¹

« L'intelligence, ce n'est pas seulement ce que mesurent les tests, c'est aussi ce qui leur échappe. »

Edgar Morin²



1. Proverbe chinois à quatre caractères, appelé *chengyu*. Un li est une unité de mesure qui équivaut à 576 mètres environ.
2. Sociologue français. Citation extraite du tome 3 de l'œuvre *La Méthode* (six tomes), intitulé *La connaissance de la connaissance* et publié en 1986.

Félicitations ! Si vous êtes toujours là après avoir lu tous les chapitres précédents, vous êtes sur la bonne voie. Mais peut-être avez-vous un peu mal à la tête ? Rassurez-vous, c'est normal.

Quand on découvre l'accessibilité web pour la première fois, son contexte d'utilisation, son cadre juridique, toutes ses règles techniques, etc., on peut observer en général deux types d'effets secondaires (outre le mal de tête) :

- une angoisse profonde associée à un sentiment d'impuissance : « Mais que n'ai-je découvert l'accessibilité web plus tôt ?! Tout ce que j'ai fait jusqu'ici ne va pas du tout ! C'est une catastrophe ! Je suis nul(le) ! Il faut que je change de métier... Bouhouhou... »
- un sentiment, au contraire, de toute-puissance, que nous appellerons le syndrome « je-sais-tout » : « Haha ! Ça ne va plus rigoler, les gogos ! Maintenant que je sais ce qu'est l'accessibilité, vous, les filous qui vous croyez tout permis en faisant n'importe quoi, eh bien moi, je vais vous apprendre de quel bois je me chauffe, vous allez voir ! »

Ces attitudes contradictoires peuvent même se succéder chez un même individu, dans un sens ou dans l'autre.

Évidemment, cette présentation est caricaturale, mais elle souligne un problème réel : comment concilier ses méthodes de travail actuelles et un objectif d'accessibilité pour tous ?

Un traitement pour ces troubles : une grande exigence, toujours, mais temporisée par de l'indulgence et de l'humilité.

Par où commencer ?

Deux axes à concilier : ce qui est faisable facilement et ce qui est crucial pour l'accessibilité

Vous êtes décidé(e) à améliorer l'accessibilité de vos pages web. Un état des lieux s'impose. Vous-même ou un intervenant extérieur réalise un audit. Le rapport permet de mettre en lumière tout ce qui ne va pas sur les pages évaluées. Et c'est qu'il y a du travail ! Alors, par où commencer ?

On peut aborder les choses de deux façons.

La première consiste à *corriger tout ce qui peut l'être facilement à moindre frais*. La tabulation n'est pas visible, ce qui empêche de suivre visuellement la navigation au clavier et invalide le critère AccessiWeb 10.7 ? Une toute petite ligne de code CSS, et c'est réparé ! Facile ! Selon les erreurs relevées, vous aurez sans doute un certain nombre

de corrections très faciles à mettre en œuvre. Commencer par ce qui est faisable est donc un axe de travail qui permet de concilier efficacité et économie. Et c'est une façon réaliste de commencer à planifier l'amélioration de l'accessibilité du site.

Mais ce n'est pas suffisant pour mettre en place une démarche accessibilité.

L'autre axe de travail, qui doit être combiné avec le précédent, concerne *les erreurs qui causent une entrave importante pour accéder aux contenus*. Vous avez un script au début de la page qui empêche vos internautes d'entrer dans le site au clavier ? Cette correction doit être prioritaire, même si c'est compliqué dans le contexte de votre page. Des informations essentielles sont contenues dans des documents en téléchargement inaccessibles ? Il faut veiller à les fournir dans un format accessible à tous. Les pages contiennent de nombreuses images d'illustration, ayant toutes un attribut `alt` n'apportant aucune information ? Il faut revoir la politique éditoriale, former les personnes en charge de mettre les contenus à jour et corriger l'alternative aux images déjà publiées pour qu'elle soit pertinente.

Le premier axe ne sera pas trop difficile à établir. Les personnes en charge de l'administration technique du site pourront vous dire assez précisément ce qui est modifiable rapidement et ce qui ne l'est pas. Par contre, le second est plus difficile à définir. Il dépendra des contenus de votre site, des erreurs trouvées et demandera une certaine expérience pour déterminer de façon pertinente les erreurs les plus gênantes pour les utilisateurs en situation de handicap.

Les niveaux A, AA et AAA des WCAG 2 ne sont malheureusement d'aucun secours pour faire ce travail d'échelonnement des réparations, le premier palier (niveau A) étant déjà extrêmement exigeant. Il manque donc des paliers pour permettre une mise en œuvre progressive de l'accessibilité.

Face à ce constat, plusieurs organismes se penchent sur la question afin de mettre en place des méthodes pour faire le tri et proposer une implémentation progressive de l'accessibilité web.

WCAG-EM : la méthodologie selon la WAI

Le W3C lui-même a pris acte de la difficulté à implémenter les WCAG 2 en l'état et travaille actuellement à une méthodologie d'évaluation de la conformité de l'accessibilité des sites web : la *Website Accessibility Conformance Evaluation Methodology 1.0*, abrégée en WCAG-EM (<http://ur1.ca/8tiua>³).

Cette méthodologie, dont le premier jet, encore sous statut de document de travail (*Working Draft*), a été publié le 27 mars 2012, est très récente et donc encore susceptible d'évoluer.

3. <http://www.w3.org/TR/WCAG-EM/>

Elle fixe pour l'heure cinq grandes étapes :

- 1 la définition de la procédure d'évaluation (champ de l'étude, objectif de l'évaluation, niveau de conformité visé...) ;
- 2 l'exploration du site pour identifier les points clés à étudier (pages importantes, fonctionnalités présentes, types de pages web, technologies utilisées) ;
- 3 la sélection d'un échantillon représentatif des contenus, fonctionnalités et technologies utilisées sur le site ;
- 4 l'audit des pages sélectionnées ;
- 5 le rapport d'évaluation.

Ce qui est assez triste avec cette méthode, c'est que non seulement elle ne répond pas aux questions que nous posons sur l'établissement de priorités parmi les critères d'accessibilité, mais en plus, elle ne fait que reproduire ce que des méthodologies d'évaluation basées sur les WCAG comme AccessiWeb, le RGAA mais également d'autres méthodes dans d'autres pays, font très bien depuis de nombreuses années. Le W3C a, en cette matière, un certain retard sur ses déclinaisons locales. Nul n'est parfait !

Heureusement, d'autres approches existent pour nous aider. Voyons ce qu'il en est en France.

Accessibility First Step : l'étape juste avant de faire de l'accessibilité

Initiée par Aurélien Lévy de la société Temesis, cette méthodologie repose sur la vérification de tests précis faisant partie d'une liste dont il suffit de suivre le déroulé. L'objectif est de permettre la vérification rapide des fondamentaux techniques de l'accessibilité. Ces fondamentaux sont déclinés en 99 tests concernant le HTML, 6 tests pour les CSS et 17 tests relatifs aux fichiers JavaScript.

Le problème souvent rencontré par les « experts » en accessibilité appelés à intervenir pour faire de l'audit, de l'accompagnement ou de la formation, est que la plupart des sites audités posent des questions qui relèvent plus d'une maîtrise de HTML et des feuilles de styles que d'un seul problème d'accessibilité.

La méthode est conçue pour être utilisée *avant* de faire appel à un consultant spécialisé en accessibilité web. L'idée consiste à dire : « Vérifiez vos fondamentaux, et si vous êtes bon, alors vous pourrez commencer à recourir à un expert ».

Attention, toutefois, comme l'écrit Élie Sloïm sur le blog de la société : « Cette liste n'a pas pour objectif de dire si votre site est accessible. Elle vous permettra de savoir si les fondamentaux de l'accessibilité sont en place. » (<http://ur1.ca/6vo1g>⁴) Elle se veut être l'étape juste avant de faire de l'accessibilité.

4. <http://blog.temesis.com/post/2011/12/15/Accessibility-first-Step>

Publiée sous la licence Creative Commons CC-BY-SA, cette *checklist* est disponible en ligne à l'adresse : <http://ur1.ca/abofy>⁵.

Elle répond en partie à la question de savoir ce qui est facilement corrigible tout de suite. Mais elle ne répond pas à la question des critères prioritaires pour éliminer les freins à l'accessibilité. Le projet MIPAW est là pour ça.

MIPAW : l'accès à l'information d'abord

MIPAW est l'acronyme pour « Méthode d'intégration progressive de l'accessibilité du Web ». Projet coordonné conjointement par l'association BrailleNet et la société Qelios, il est strictement fondé sur les critères développés dans les WCAG 2, mais redéfinit un ordre de priorités en fonction d'un axe : l'accès à l'information.

Présenté au séminaire AccessiWeb du 18 janvier 2012, ce projet MIPAW est toujours en cours d'élaboration.

Outre l'axe principal d'accès à l'information, il distingue quatre étapes progressives, les deux premières étapes se plaçant à gauche de l'axe central « accès à l'information », et les deux dernières à droite (donc moins prioritaires pour l'accessibilité) :

- 1 sécuriser l'accès à l'information (accès technique) ;
- 2 garantir l'accès à l'information (accès de l'utilisateur) ;
- 3 améliorer l'impact utilisateur ;
- 4 améliorer l'expérience utilisateur.

L'objectif du projet, à terme, est de fournir une liste de critères répartis selon ces quatre étapes, indépendamment du niveau Bronze, Argent ou Or.

Pour qu'un site permette réellement l'accès à tous les internautes, il faudra donc respecter les critères situés en priorité à gauche de l'axe, dans les deux premières étapes. Ce travail devrait permettre de répondre à la question des critères à traiter en priorité pour éviter de freiner les utilisateurs dans leur accès au site.

Nous aurons donc ainsi des listes nous permettant de savoir :

- quels sont les critères faciles à implémenter tout de suite (avec Accessibility First Step) ;
- quels sont les critères prioritaires pour l'accès au site (avec MIPAW).

C'est bien joli tout ça, mais comment répartir ces listes et ce travail au sein de l'équipe projet et des différents intervenants responsables du site ? Le projet AcceDe Web donne des pistes de solutions.

5. <http://checklists.opquast.com/fr/accessibilityfirststep>

AcceDe Web : l'accessibilité selon son profil et selon la phase projet

Le projet AcceDe Web, piloté par la société Atalan, consiste à proposer des notices pratiques en fonction de la tâche à accomplir, et donc du profil de l'intervenant.

Quatre types de profils sont dégagés, avec une notice différenciée par profil :

- pour les graphistes/maquettistes, une notice d'accessibilité pour la conception graphique ;
- pour les intégrateurs et les développeurs :
 - une notice d'accessibilité HTML/CSS,
 - une autre notice dédiée à l'accessibilité des interfaces riches et à JavaScript ;
- pour les contributeurs éditoriaux : un modèle de document pour concevoir une notice d'accessibilité éditoriale.

Les notices AcceDe Web sont disponibles en ligne sous licence Creative Commons BY 3.0 à l'adresse : <http://accede-web.com>.

Quelles que soient vos habitudes en matière de gestion de projet, retenez qu'il n'existe pas une gestion de projet accessible. L'accessibilité ne constitue pas une étape à part dans le processus de création ou de refonte d'un site, mais plutôt un élément transversal qui peut (doit !) s'intégrer, quelles que soient vos méthodes de gestion de projet.

Les méthodes et les listes proposées par différents organismes, souvent sous licence libre, offrent une aide dont vous pouvez vous servir selon vos besoins. Mais elles ne servent à rien si vous n'entamez pas vraiment une démarche accessibilité.

« Et l'accessibilité ? » : une démarche en une question

Vous allez peut-être trouver notre propos simpliste, pourtant, mettre en œuvre une démarche en faveur de l'accessibilité web repose sur une toute petite question, la même question qui doit revenir tout le temps : « Et l'accessibilité ? »

Vous discutez avec des collègues à la machine à café, le responsable évoque sa volonté de refaire le site de l'entreprise pour le rafraîchir, lui donner un coup de jeune et mieux coller aux nouveaux objectifs de la société. Et là, vous commencez tout en douceur votre travail de sensibilisation par cette simple question : « Et l'accessibilité ? »

Il y a fort à parier que tout le monde va se tourner vers vous et vous répondre quelque chose comme : « Gnnnin ?... »

À ce moment là, c'est à vous de jouer. Parez-vous de votre plus beau sourire, et commencez avec patience, pédagogie et passion, à expliquer ce que c'est et en quoi c'est important.

Dès le moment où l'idée d'une création ou d'une refonte de site germe, il faut tout de suite amener le sujet de l'accessibilité. Plus la question est prise tôt, plus il y a de chances pour que vous parveniez à produire un résultat à la hauteur de vos attentes... et à moindre coût.

À SAVOIR Coûts et bénéfices de l'accessibilité du Web

L'accessibilité a de nombreux avantages indirects : maintenance, référencement, respect des standards, qualité... Il aurait fallu tout un chapitre pour les détailler tous.

Le coût de l'accessibilité web est dérisoire par rapport au coût de mise en accessibilité d'un bâtiment, mais sur le principe, on peut retenir que plus l'accessibilité a été prise en compte tôt (dès les plans de l'architecte), moins c'est coûteux.

D'un point de vue opérationnel, le budget consacré à l'accessibilité sera établi à partir du coût de formation des équipes, et éventuellement du coût d'un accompagnement ou d'un audit extérieur. Par contre, le budget pour la réalisation proprement dite ne sera pas franchement différent pour un site accessible que pour un site qui ne l'est pas, les coûts étant dilués et non perceptibles.

Mais outre les bénéfices et les coûts de l'accessibilité, n'oubliez jamais de vous poser la question suivante : quel serait le coût de la non-mise en accessibilité d'un site ? Entre perte d'utilisateurs et risques juridiques, le calcul devrait être vite établi.

Autre exemple : vous êtes une collectivité publique et vous devez réaliser un appel d'offres pour refaire le site. On vous donne des consignes pour la rédaction et le service communication fixe les grands objectifs... Et là, vous posez la question naïvement : « Et l'accessibilité ? »

Fixer des critères de notation concernant l'accessibilité dans le cahier des charges permettra de donner du poids à cette question dans le choix du prestataire et dans la suite du déroulé de l'opération.

Cette simple question en entraîne donc beaucoup d'autres : qu'est-ce que l'accessibilité ? Quel poids accorder aux compétences en accessibilité dans le choix du prestataire ? Quelle stratégie de communication mettre en place pour n'exclure personne ? Faut-il avoir recours à un consultant externe pour donner un avis objectif sur le travail des prestataires ou se former en interne pour le faire nous-mêmes ? En tant qu'entreprise, dois-je faire le choix de former mes collaborateurs ? Vaut-il mieux sous-traiter ? Comment nous positionner sur cette question quand nous répondons à un appel d'offres ? Et bien d'autres questions encore...

Ouvrir les vannes des questions sur l'accessibilité : voilà en quoi consiste la mise en place d'une démarche en faveur de l'accessibilité. Tous les arbitrages n'iront pas nécessairement dans le sens du respect de tous les critères, mais au moins des choix auront-ils été faits, en conscience, et il ne sera plus possible de reporter la responsabilité de l'inaccessibilité du site sur une méconnaissance.

DÉBAT Vous avez dit « surqualité » ?

Nous abordons là un sujet qui pourrait ne pas poser question s'il ne servait d'alibi à certains pour ne rien faire. Il s'agit du problème de la surqualité.

Cette question n'est pas spécifique à l'accessibilité, mais se pose dans tous les domaines concernés par la gestion de la qualité. On pourrait la résumer par l'adage « le mieux est l'ennemi du bien ». Il s'agit en effet de veiller à ce que le niveau de qualité optimal ne produise pas de coût inadéquat. C'est ce surcoût par rapport au bénéfice attendu qui est appelé « surqualité ».

La mise en garde s'adresse donc en priorité aux personnes fraîchement formées et qui souffrent encore du syndrome « je-sais-tout » que nous décrivions en introduction du chapitre. Le risque à vouloir suivre les règles à tout prix sans prendre en compte le contexte du projet serait donc de deux ordres :

- soit se priver de faire des pages web belles et dynamiques pour faciliter le respect des règles, ce qui assimilerait négativement l'accessibilité à un Web pauvre et dépassé ;
- soit se comporter comme un dictateur tyrannique refusant de comprendre les réalités et complexités techniques du projet, en faisant alors passer l'accessibilité pour quelque chose de nuisible (les experts accessibilité, empêcheurs de tourner en rond).

Ces risques posés par la surqualité sont réels. Mais avant de crier au loup, regardons la réalité de la situation de l'accessibilité aujourd'hui : la *très très* grande majorité des sites web en France ne respectent pas les principes de base de l'accessibilité. Nous n'avons pas d'étude précise sur laquelle nous appuyer, mais dire que 90 % des sites ne sont pas accessibles ne doit pas être loin de la vérité.

Manque de formation, de connaissance du sujet... Plusieurs facteurs peuvent expliquer ce problème. Mais parmi ces raisons, le manque de volonté existe. Or, nous avons pu constater une reprise de l'argument « surqualité », dévoyé, mal compris, pour justifier de ne rien faire.

Voici le genre de propos entendu en réunion client : « Du sous-titrage ? Non, ne tombons pas dans la surqualité. En plus c'est du AA. » Les personnes sourdes et malentendantes apprécieront.

Au lieu de faire passer pour intégristes des personnes vous proposant de trouver des solutions pour permettre l'accès aux contenus à tous, sans discrimination, il est temps de remettre la balance dans le bon sens. Un site inaccessible est discriminatoire. Refuser de prendre en compte un problème réel d'accessibilité en se cachant derrière un argument de surqualité n'est pas acceptable.

Bien sûr, selon le budget, les délais et les volontés des uns et des autres, il ne sera pas toujours possible de prendre en compte tous les critères d'accessibilité. Mais cela doit se faire en connaissance de cause. Vous ne voulez pas de sous-titrage ? Très bien. Assumez dans ce cas que vos vidéos ne seront pas accessibles aux personnes sourdes et malentendantes. C'est un choix, mais certainement pas un problème lié à l'accessibilité web elle-même.

Une démarche progressive et continue : accessibilité et conduite du changement

Vous avez ouvert les vannes des questions sur l'accessibilité ? Bravo ! Vous êtes dans une démarche d'accessibilité. Mais attention ! Croire qu'il suffit de répondre à toutes ces questions à un instant t pour avoir un site accessible une bonne fois pour toute est une erreur. La mise en accessibilité est une démarche progressive et continue.

Se former, sensibiliser, convaincre

Prendre en compte l'accessibilité ne va pas de soi. C'est une vraie démarche que l'on met en place après avoir été soi-même sensibilisé. Peut-être la lecture de ce livre vous donnera-t-elle envie de vous y essayer. Nous l'espérons, en tous cas. Cependant, un site est rarement l'œuvre d'une seule personne. C'est le fruit d'un travail collectif, et pour prendre en compte l'accessibilité, il va d'abord falloir commencer par convaincre votre entourage de s'intéresser au sujet.

Si vous arrivez au terme de la lecture de ce livre, vous avez déjà fait un bout du chemin. Vous êtes vous-mêmes sensibilisé. Selon la rigueur avec laquelle vous aurez lu les chapitres dédiés aux critères d'accessibilité, vous serez peut-être même formé.

Pour intéresser vos collègues, le mieux est sans doute d'organiser un petit-déjeuner (ou un goûter, ou autre chose) de sensibilisation. Susciter l'intérêt, attiser la curiosité, mettre en avant les avantages, prévenir des risques juridiques... autant d'arguments que vous aurez à déployer pour convaincre. Selon le contexte et l'accueil reçu par votre entourage, vous devrez affiner vos arguments. Retenez que le sourire et la bonne humeur sont généralement plus efficaces que la culpabilisation et la menace.

Une fois acté le principe de prendre en compte l'accessibilité dans le projet, il va falloir former l'ensemble de vos collègues. Des changements dans les méthodes de travail devront peut-être intervenir. Votre intégrateur ne jure que par les balises de mise en forme et ne connaît pas bien les CSS ? Il va sans doute falloir lui suggérer de se mettre à jour, au risque de mettre en échec tous les efforts déployés pour l'accessibilité.

Vos contributeurs adorent les liens « cliquez ici » et préfèrent joindre un document PDF plutôt que d'entrer le contenu dans la page web ? Outre une formation spécifique, c'est tout un changement d'habitudes qu'il va falloir faire accepter. Et c'est loin d'être facile, surtout si on ne comprend pas l'utilité de ces changements. La sensibilisation au handicap est donc une première étape indispensable.

RESSOURCE Ne pas changer, c'est tourner en rond

Nous vous recommandons la lecture d'un article de Al Sparber de juin 2003. Il prend une métaphore pour évoquer les difficultés et les résistances au changement. Cet article a été traduit en français et il est disponible sur le site de Pompage :

► <http://ur1.ca/abpc9^a>

L'auteur raconte qu'il revient un jour avec une surprise pour ses enfants : un poney. Clarence (c'est le nom du poney) a été élevé dans un manège : il a appris à tourner en rond. Or, quand les enfants essaient de monter le poney pour faire un tour dans le jardin, Clarence ne sait rien faire d'autre que de tourner en rond.

Et de conclure : « La conception de pages web, c'est comme la médecine. Les médecins prospères sont ceux qui se tiennent au courant des dernières techniques. Personne ne souhaite avoir recours à un docteur qui ignorerait tout de l'imagerie médicale et de la chirurgie non intrusive. Je n'embaucherais pas un webdesigner que CSS, les standards du W3C, l'accessibilité et l'ergonomie ne mettent pas d'humeur lyrique. »

Finalement, à force de patience, les enfants ont réussi à changer les habitudes du poney Clarence. Persévérez, vous allez y arriver !

a. <http://www.pompage.net/traduction/clarence>

Une amélioration continue : « tendre vers » l'accessibilité

Une idée reçue persistante consiste à penser qu'on peut créer des sites 100 % accessibles. Voulant bien faire, certains se fixent d'ailleurs cet objectif pour leur propre site. Méfiez-vous de ceux qui vous offriront leurs services pour réussir ce genre d'exploit. Car non seulement c'est impossible, mais ce n'est pas non plus toujours souhaitable.

Dans l'introduction des WCAG 2, nous pouvons lire : « Notons que même du contenu conforme au niveau le plus élevé (AAA) ne sera pas accessible aux personnes ayant tous les types, les degrés ou les combinaisons de limitations fonctionnelles, en particulier dans les domaines cognitifs, du langage et de l'apprentissage. » (<http://ur1.ca/abp64⁶>)

Il vous suffit de relire la première partie de cet ouvrage pour vous rendre compte que nous ne pourrions jamais atteindre la perfection d'un site accessible à tous. Ce qu'il est possible de faire, en revanche, c'est de *respecter 100 % d'une norme*. Et encore, si vous avez lu les chapitres précédents, vous vous serez rendu compte que cela reste très difficile.

Faire de l'accessibilité, ce n'est pas viser l'impossible, mais faire de son mieux pour améliorer l'accès au site en prenant en compte tous les critères possibles. Il est important d'aborder cette question avec humilité, tant la tâche est ardue.

Le risque en se fixant des objectifs inatteignables, c'est de vous décourager et de ne plus rien faire, ou de vous contenter d'un minimum qui sera très insuffisant pour garantir l'accès des personnes en situation de handicap à vos contenus web. C'est pour éviter cet écueil que les WCAG recommandent de ne pas viser un niveau AAA à tout prix.

6. <http://www.w3.org/Translations/WCAG20-fr/#intro-layers-guidance>

Soyez ambitieux mais réaliste, exigeant mais indulgent ! Certains critères AccessiWeb peuvent paraître très simples à mettre en place. Cela est souvent le cas sur un site statique réalisé par un professionnel. Mais la production des pages HTML est rarement faite de façon statique, et les personnes qui mettent à jour le contenu ne sont pas toutes des professionnelles du Web. Selon le logiciel (CMS) utilisé, le nombre de personnes intervenant sur le site (et qu'il faut donc former) et leur profil technique, ou encore les conditions du projet (délais, budget, fonctionnalités), il peut être plus difficile de respecter certains critères qui semblent pourtant faciles de prime abord.

L'accessibilité n'est pas une fonctionnalité. On la rencontrera tout au long du projet sous la forme de points à vérifier, à différentes étapes, par différentes personnes. À l'instar de la sécurité ou du référencement, elle est à prendre en compte dès le début du projet, et à suivre de manière simple mais rigoureuse tout au long de la vie du site.

Plus que le niveau effectivement atteint à la livraison, ce qui compte, c'est la mise en œuvre de la démarche, la définition de la direction à suivre. Dans tous les cas, n'attendez pas qu'un site remplisse 100 % de votre objectif pour communiquer sur votre démarche.

L'accessibilité... accessible à tous ?

Nous écrivions en introduction que ce livre s'adressait à tout le monde et que chacun pouvait contribuer à l'accessibilité web. Pourtant, en lisant ce livre, vous avez également découvert que l'accessibilité web était un domaine de compétences à part entière, ouvrant la porte à de vrais métiers, bien que méconnus.

Les métiers de l'accessibilité du Web : de la formation à la professionnalisation

Si le terme « expert » est accolé au moindre des nouveaux métiers du Web, les formations Expert AccessiWeb en Évaluation (abrégié en EAE) ont ouvert la voie depuis 2003 à la constitution d'une communauté de professionnels formés à l'accessibilité. Aujourd'hui, les moyens de se former à l'accessibilité se sont multipliés et la formation dispensée par BrailleNet n'est plus le seul moyen d'acquérir une expertise. Une professionnalisation des « experts », spécialisés selon leurs compétences, a permis l'émergence de nouveaux métiers, tout en maintenant l'exigence d'une connaissance de base de l'accessibilité pour tous.

Spécialisation et transversalité

Intégrateur, développeur, graphiste, ergonomiste, expert SEO... Autant de métiers avec leurs particularités et leur domaine d'expertise. Pourtant, tous ces professionnels agissent sur la création d'un site web et comme nous l'avons vu, il n'est possible de rendre un site accessible qu'à condition que tout le monde prenne sa part dans le processus. Une culture commune est donc nécessaire à ce sujet. Et à lire les différents livres publiés sur ces sujets pourtant variés, on y voit l'accessibilité très souvent abordée, démontrant qu'il s'agit bien d'une préoccupation pour beaucoup.

Il est donc nécessaire de connaître son métier, mais aussi d'être sensibilisé à l'accessibilité.

Cela dit, au-delà d'une sensibilisation de tous les acteurs de la chaîne de production web, il peut être utile de recourir aux compétences d'un professionnel de l'accessibilité web dont c'est la spécialité.

Les métiers dédiés à l'accessibilité : évaluateur, consultant, formateur

La première question que nous pouvons nous poser est : le métier d'expert en accessibilité existe-t-il ? La réponse sera simple : non. Pas à nos yeux. Le mot lui-même « d'expert » est tellement dévoyé, utilisé à tort et à travers que c'en est presque devenu un argument commercial pour faire valoir qu'on est le meilleur. Impossible, dès lors, de ne pas se dire « expert » au risque de passer pour moins compétent que son concurrent.

Pourtant, l'expert devrait être celui qui pratique une expertise, c'est-à-dire un professionnel qui évalue l'état d'un site en fonction d'un référentiel. L'expert qui fait votre état des lieux de sortie quand vous quittez un logement fait donc une expertise.

Compte tenu du contexte qui est celui du Web et des pratiques que nous venons de décrire, nous emploierons plutôt le terme d'*évaluateur*. L'évaluateur en accessibilité va étudier, critère après critère, si la page est conforme au référentiel employé (AccessiWeb, RGAA, WCAG 2...). Le niveau d'expertise est le moins élevé parmi les professions en accessibilité. C'est le niveau de base.

Puis vient le *consultant* en accessibilité. En plus de faire des évaluations, le consultant bénéficie d'une expérience qui lui permet de proposer des conseils et des solutions dans des cas complexes. Ses qualités humaines, pour être capable de comprendre les problématiques du client et de synthétiser les solutions qu'il propose dans un langage intelligible sans connaissances techniques, sont aussi importantes que ses connaissances approfondies de l'accessibilité web.

Par ailleurs, vous trouverez également le métier de *formateur* en accessibilité. En plus de connaître sa matière, un formateur doit faire preuve de pédagogie, savoir élaborer un plan de formation en adéquation avec les objectifs attendus, rédiger un programme tenant compte des attentes et des horaires, etc.

Mais au-delà de ces trois métiers spécifiques à l'accessibilité, et qui nécessitent tous une parfaite connaissance des règles d'accessibilité et l'adoption d'une méthodologie, il n'est nul besoin d'être expert.

IDÉE REÇUE « Ben, maintenant que j'ai lu le livre, plus besoin d'avoir recours à un professionnel alors ! »

Notre propos n'est pas de dire que les professionnels ne servent à rien. Mais plutôt que la seule expertise est très insuffisante pour faire de l'accessibilité. Croire qu'il suffit de déléguer la question ou d'appuyer sur un bouton pour faire accessible, c'est aller tout droit vers un échec assuré. Ce livre a pour objectif de combler des lacunes et de permettre à chacun d'évaluer ce que représente l'accessibilité web en termes d'enjeux, de connaissances à acquérir, d'efforts à fournir... Pour que chacun, expert ou non, puisse trouver sa place et prendre sa part dans la mise en place d'une démarche en faveur de plus d'accessibilité web. Selon vos objectifs et vos moyens, il peut être judicieux de recourir à un professionnel. Car si on peut faire de l'accessibilité sans être expert, cela risque de prendre du temps (et donc de l'argent) et d'être aussi laborieux qu'imparfait. Sans expertise, vous ne pourrez faire que « *ce que vous pouvez* », bien accompagné, à charge de travail et à coût égal, vous ferez « *ce qu'il faut* ».

La spécialisation nécessaire des intégrateurs et des contributeurs

S'il est vrai que toute personne amenée à travailler sur le Web devrait être sensibilisée à l'accessibilité pour optimiser les chances de succès du projet, il y a deux métiers qui exigent un peu plus qu'une sensibilisation tant leur travail est indispensable pour la mise en place de l'accessibilité : ce sont les intégrateurs et les contributeurs.

Sans devoir en faire des experts, une formation spécifique sera indispensable pour l'application concrète des règles et un maintien de l'accessibilité dans le temps.

Ce sont les *intégrateurs* (développeurs front-end) qui transforment les images remises par le graphiste en code HTML. Or, une grande partie des critères d'accessibilité trouvent leur application dans la conception du gabarit HTML. Il est donc essentiel que les intégrateurs aient une bonne connaissance des règles qu'ils devront s'attacher à respecter pour prendre l'accessibilité en compte dès le départ.

Puis, une fois le site en place, il va vivre, évoluer avec un enrichissement et une mise à jour constante des contenus. C'est aux *contributeurs* que revient cette tâche. Le profil technique de ces contributeurs peut être très varié. Certains ont de grandes qualités d'écriture et d'animation, mais aucune connaissance technique du Web. Seule une formation spécifique peut permettre un maintien de l'accessibilité dans le temps. Il ne s'agit pas de leur expliquer toutes les règles d'accessibilité, ni d'en faire de fins connaisseurs de HTML. Mais l'étude de certaines règles d'accessibilité dédiées à la contribution, avec une mise en pratique dans leur contexte de production (une bonne utilisation du CMS, par exemple), est indispensable.

Tout le monde peut contribuer !

Que vous souhaitiez devenir un professionnel de l'accessibilité ou que vous vous intéressiez à la question « en passant », sans avoir de responsabilité technique, vous pouvez contribuer à rendre les sites web plus accessibles.

Comme nous l'avons répété tout au long de cet ouvrage, pour faire de l'accessibilité, il faut déjà y penser. Et encore de trop nombreuses personnes ignorent jusqu'à l'existence de cette discipline.

Vous pouvez essayer de sensibiliser votre entourage professionnel, en commençant par leur expliquer ce que vous aurez retenu de ce livre. Si vous êtes un professionnel du Web, vous pouvez essayer de mettre en pratique certaines des règles d'accessibilité que nous avons abordées.

Le seul mot d'ordre sera : faites de votre mieux. Ce ne sera peut-être pas complet, ce sera sans doute imparfait à la première tentative, mais qu'importe : essayez !

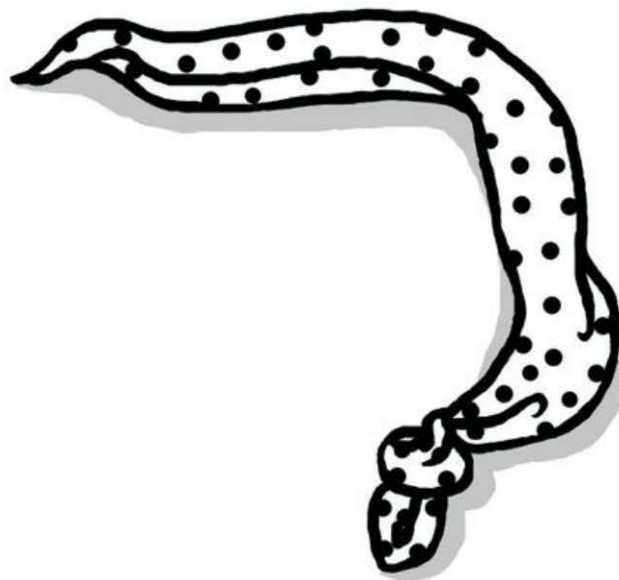
La volonté de trop bien faire conduit parfois à l'inertie. Or, si l'erreur est humaine, ne rien tenter en sachant les conséquences pour les utilisateurs serait coupable. Alors faites preuve d'indulgence, tant envers vous-même qu'envers les autres. Changer ses habitudes n'est pas facile. Et tout le monde fait des erreurs (l'auteur de ce livre n'échappe pas à la règle !)

CONCLUSION

Accessibilité et liberté

« Les Hommes naissent et demeurent libres et égaux en droit »

Article 1 de la Déclaration des droits de l'Homme et du citoyen, 1789



L'illustration en avant-propos figurait un personnage assis devant son clavier, les mains attachées dans le dos. Ce dessin est issu d'une campagne menée par le groupe de travail « Accessibilité et logiciels libres » de l'association April pour sensibiliser à l'accessibilité numérique. Une affiche a été réalisée avec ce dessin, introduit par la phrase suivante : « Comment écririez-vous un message si... » puis le dessin du personnage les mains attachées dans le dos¹.

L'illustration de la conclusion répond à ce premier dessin : on y voit le foulard qui sert à entraver le personnage flotter au vent.

Rendre une page web accessible, cela revient à détacher ce foulard pour les personnes en situation de handicap, à les *libérer*.

Quand on a un handicap permanent, quel qu'il soit, on est très souvent entravé, empêché de réaliser des choses très simples à cause de lui, mais également en raison d'un environnement non accessible. On peut alors se sentir parfois prisonnier, prisonnier d'un corps qui ne fonctionne pas comme on voudrait, prisonnier dans un environnement qui n'est pas pensé pour nous accueillir, privé très injustement d'une liberté qui ne devrait même pas poser question.

Évidemment, tout le monde est concerné par l'accessibilité, car tout le monde est concerné par le vieillissement, et on peut tous être en situation de handicap de façon provisoire au cours de sa vie. *L'accessibilité bénéficie donc à tous.*

Mais pour certaines personnes, cette situation n'est pas provisoire et le mot qui revient le plus souvent quand on discute du retentissement du handicap dans la vie quotidienne est « parcours du combattant ».

Quand on découvre son handicap après une vie valide, il y a toujours un « avant » et un « après ». Les premiers temps, on refuse de croire ce qui nous arrive, et on lutte pour retrouver sa vie d'avant. Quand on voit que c'est impossible, on est alors pris d'une violente colère contre cette injustice et contre tout cet environnement qui ne comprend pas et « ne fait aucun effort ». Car c'est bien vrai, il est impossible de réellement comprendre ce qu'implique un handicap dans une vie quand on ne l'a pas vécu soi-même. Et selon la personne, cela peut être vécu très différemment.

Puis à la colère succède la résignation et une profonde tristesse : toutes ces choses que l'on ne pourra plus jamais faire, tout ce qu'on a perdu, tout ce qu'on aurait pu faire et qu'on ne réalisera jamais. Détresse et frustration. Un passage difficile mais incontournable et nécessaire pour faire le deuil de sa vie « d'avant ».

Une fois surmontée cette période, qui peut durer des années parfois et que certains peuvent d'ailleurs ne pas surmonter, une nouvelle vie s'ouvre. Celle de « l'après ».

1. Affiches libres à télécharger sur le site de l'April :
<http://www.april.org/deux-affiches-pour-sensibiliser-laccessibilite>

Une vie pas moins riche, pas moins heureuse, mais différente. Le temps est venu de l'autodérision pour ne pas susciter la pitié (n'avez-vous jamais remarqué l'humour de beaucoup de ces personnes handicapées qui comptent peut-être parmi vos amis ?), et aussi du système D pour survivre dans un monde hostile, et particulièrement inaccessible en France.

Le Web est une bouffée d'air dans cet univers. Alors que la moindre sortie comporte des marches, des passages trop étroits, des gens qui parlent dans leur barbe sur fond de musique dans les supermarchés, le manque de patience des personnes pressées de rentrer chez elles et qui vous bousculent, la difficulté de communiquer pour certains avec des personnes non formées... Le Web est un espace douillet sur lequel on se rend depuis chez soi, confortablement installé et sans craindre les regards curieux. Un espace où l'on peut faire ses courses, discuter avec des amis, s'informer... Bref, un espace de liberté retrouvée, où on peut enfin se sentir « comme tout le monde ».

Mais pour que cette liberté soit une réalité, il vous revient, cher lecteur, d'œuvrer pour que les sites dont vous avez la charge soient accessibles.

Nous avons vu combien il était souvent ridiculement simple d'un point de vue technique de prendre en compte l'accessibilité. Une alternative pertinente à une image, des étiquettes sur les formulaires, un balisage HTML correct de la page, une structuration logique... En dehors des cas relativement rares où il peut être nécessaire de développer des solutions complexes pour rendre la page accessible, l'accessibilité tient d'abord au fait d'y penser.

Vous avez certes déjà à penser à beaucoup de choses : qualité, performance, sécurité, référencement... le tout dans des délais souvent très courts. Mais si toutes ces composantes sont importantes à prendre en compte, la priorité doit être donnée à l'accessibilité. C'est une question de droits, une question de justice, une question de liberté.

Aucune loi ne vous imposera de réaliser des sites bien positionnés dans les moteurs de recherche, mais la loi vous demande de ne pas discriminer vos internautes.

Ce sont de petits gestes qui ne coûtent pas grand chose, mais qui permettent de rendre leur liberté à certaines personnes pour qui le Web est vital.

Si vous considériez votre travail comme un simple empilement de lignes de codes, vous ne liriez sans doute pas ce livre. En tant que développeur, concepteur de site, graphiste web, rédacteur, commercial, décideur : vous avez le pouvoir de donner plus de liberté à des millions d'internautes.

La seule question à vous poser est la suivante : Êtes-vous prêt(e) à changer le Web ?

ACTION Accessibilité et logiciels libres

Un logiciel libre répond à quatre libertés :

- la liberté d'utiliser le logiciel, sans contrainte ;
- la liberté d'étudier son mode de fonctionnement (ses sources) ;
- la liberté de modifier le logiciel ;
- la liberté de partager le logiciel, légalement, librement, sans se voir taxé de « pirate ».

Bien qu'on puisse trouver gratuitement en téléchargement la plupart des logiciels libres, la question de la liberté prédomine donc sur celle de la gratuité. Et c'est en cela qu'ils offrent une solution pour permettre, à terme, une plus grande accessibilité pour tous, et donc plus de liberté.

Un logiciel libre est-il plus accessible ?

Non, pas forcément. De même qu'il n'est pas forcément plus performant ni meilleur d'un point de vue technique. Mais sa licence d'utilisation permet d'apporter si besoin les changements nécessaires pour le rendre plus accessible. Quitte à payer des développeurs compétents pour faire cette tâche à votre place. Car liberté n'est pas synonyme de gratuité, et un système économique du Libre existe, s'appuyant sur la vente de services (développement, formation, maintenance).

Pourtant, le monde du handicap constitue un marché extrêmement fermé où quelques rares sociétés gardent jalousement leur secret de fabrication pour vendre à des prix exorbitants des logiciels à la qualité parfois discutable. Le pire est que ces logiciels se voient subventionnés par de l'argent public quand des solutions libres de qualité existent.

Libérer les internautes, c'est aussi contribuer à sortir de ce système fermé pour rendre le choix des outils aux utilisateurs, en toute connaissance de cause.

Pour en savoir plus sur ce sujet, vous pouvez consulter le cahier accessibilité de l'opération candidats.fr sur le site de l'April :

▶ <http://ur1.ca/abkfx^a>

Pour découvrir le logiciel libre en action, rendez-vous sur l'annuaire francophone de Framasoft, à partir duquel vous pourrez télécharger des centaines de logiciels pour les tester ou les adopter :

▶ <http://www.framasoft.net>

Enfin, si vous êtes intéressé/e pour contribuer sur ce sujet, rendez-vous sur :

▶ <http://libre-et-accessible.org>

a. http://wiki.april.org/w/Cahier_accessibilit%C3%A9_pour_candidats.fr

ANNEXE

Définition des abréviations par ordre alphabétique

CMS : de l'anglais *Content Management System*, il s'agit d'un logiciel pour gérer les contenus d'un site web.

DGME : Direction générale de la modernisation de l'État.

DISIC : Direction interministérielle des systèmes d'information et de la communication.

DTD : déclaration de type de document (le type de document est abrégé en « Doctype »).

EAE : Expert AccessiWeb en Évaluation (fait référence à la formation dispensée par l'association BrailleNet).

GTA : Groupe de travail AccessiWeb.

LCEN : Loi sur la confiance dans l'économie numérique.

MIPAW : Méthode d'intégration progressive de l'accessibilité du Web.

RGAA : Référentiel général d'accessibilité des administrations.

W3C : *World Wide Web Consortium*.

WAI : *Web Accessibility Initiative*, groupe de travail sur l'accessibilité du Web au sein du W3C.

WCAG : se prononce « ouécague ». Vient de l'anglais *Web Content Accessibility Guidelines* qu'on traduit par « règles d'accessibilité des contenus web édictées par la WAI ».

WHATWG : *Web Hypertext Application Technology Working Group*, groupe de travail regroupant les développeurs de navigateurs pour travailler au développement de nouvelles technologies destinées à faciliter l'écriture et le déploiement d'applications à travers le Web.

Index

A

- a11y 13
- abréviation 202
- AcceDe Web 312
- accessibilité
 - conduite du changement 315
 - coûts et bénéfices 313
 - définition du WAI 9
 - démarche 307, 312
 - formation 315
 - métiers 317
 - sensibilisation 315
- Accessibility First Step 310
- AccessiWeb 73, 80, 85
 - Cadres 105
 - Consultation 256
 - Couleurs 107
 - critères 75, 87
 - Éléments obligatoires 183
 - Expert en Évaluation 73, 317
 - Formulaires 229
 - HTML5 272
 - Images 89
 - Liens 157
 - Navigation 251
 - Présentation de l'information 207
 - Scripts 166
 - Structuration de l'information 195
 - Tableaux 133
 - thématiques 75, 87
- acronyme 202
- agent utilisateur 28, 190
- Ajax (Asynchronous JavaScript and XML) 41, 167
- Alsacrérations 40

- amélioration
 - continue 316
 - progressive 172
- application 281
- ARIA (Accessible Rich Internet Applications Suite) 282
 - rôles 284
- attribut HTML 184
 - abbr 141
 - alt 90, 112, 163, 279
 - ARIA 283
 - axis 155
 - controls 280
 - for 230
 - id 151, 230
 - interdit 208
 - longdesc 99
 - role 284
 - scope 140
 - summary 137
 - tabindex 173
 - title 105, 112, 158, 165, 231, 240, 259
 - wmode 131
- audiodescription 126, 280
- audit d'accessibilité web 70

B

- balise HTML 184
 - <a> 158, 175, 302
 - <abbr> 203
 - <applet> 90
 - <area> 90
 - <article> 275
 - <aside> 275
 - <audio> 280
 - 209

<button> 175, 236
 <caption> 136
 <dd> 266
 <dfn> 266
 <div> 274
 <dl> 199, 266
 <dt> 266
 209
 <embed> 281
 <fieldset> 240–241
 <figcaption> 278
 <figure> 278
 <footer> 275
 <frame> 105
 <h1> 196, 277
 <h2> 196
 <header> 275
 <i> 209
 <iframe> 105
 90
 <input> 90, 175, 230, 236, 279
 <label> 230
 <legend> 240
 <meter> 281
 <nav> 275
 <noscript> 171
 199
 <optgroup> 244
 <progress> 281
 <q> 204
 <section> 275
 <select> 175, 230, 242
 209
 <style> 210
 <table> 137, 191
 <td> 136
 <textarea> 175, 230
 <th> 136
 <title> 189
 <track> 280
 199
 <video> 280
 état 292
 interdite 208

propriétés 292
 BrailleNet 73
 breadcrumbs 252

C

cadre 105
 attribut title 105
 copier-coller 107
 captcha 23, 99
 Cercle Chromatique Accessible 117
 citation 204
 clavier virtuel 33
 Color Contrast Check 116
 Colorblind Web Page Filter 111
 ColorZilla 221
 composant d'interface 286, 298
 conduite du changement 315
 consultation 256
 contenus en mouvement ou clignotants 268
 documents en téléchargement 262
 émoticônes 266
 flashes 268
 jargon 266
 maîtrise de l'environnement 257
 prononciation 267
 contexte
 changements 176
 contraste 113
 couleur 107
 alternative 108
 contraste 113
 de fond 214
 héritage 215
 CSS (Cascading Style Sheet)
 cascade 215
 désactivation 210

D

Daisy Talking Book 4
 Dasher 33
 déficience 16
 dégradation élégante 172
 démarche 307
 Design for All 14

Doctype 186
droit d'auteur 132
DTD (définition du type de document) 186
dyslexie 225

E

élément
 obligatoire 183
 propriétaire 185
environnement maîtrisé 173
épilepsie 268
 photosensible 23
ergonomie 37
évaluation de l'accessibilité dun site 70

F

feuille de styles 208
fil d'Ariane 228, 252
Firebug 215, 268
Flash 127, 131, 280
formulaire 229, 279
 erreurs 245
 étiquette 230
 intitulé des boutons 236
 libellé des champs 230
 listes de choix 242
 saisie des données 245
 structuration 240

G

glisser-déposer 296
GTA (Groupe de travail AccessiWeb) 73

H

handicap 10, 12, 16
 accès au multimédia 119
 auditif 20, 35, 122
 invisible 25
 mental 22
 moteur 21
 visuel 18
HeadingsMap 197
headstick 33
HTML 183
 abréviation 203

ARIA 283
 citation 204
 composants d'interface 287
 drag&drop 296
 formulaires 279
 images 278
 interactivité 281
 langue du document 188, 191
 liste 199
 multimédia 279
 relations entre éléments 297
 sens de lecture 194
 spécification 184
 structuration 195, 274
 structure 286
 tableau de présentation 191
 titre 196
 titres 277
 widget 298
HTML5 88, 271–272
 formulaires 250
 liens 302

I

IAN (Institut de l'accessibilité numérique) 73
image 89
 alternative 96
 attribut alt 90
 captcha 99
 crédits 93
 d'illustration 91
 de décoration 91
 désactivation 210
 légende 97, 278
 lien 162
 porteuse d'information 91
 texte 103
incapacité 16
interactivité 281

J

JavaScript 41, 166, 281, 301
 alertes non sollicitées 181
 alternative 167

- bibliothèques 167
- changements 176
- contrôle 174
- raccourcis clavier 174
- test 171

Jaws 28

Juicy Studio 221

L

landmark role 254, 284

lecteur multimédia 127

lecture d'écran 28

lien 157

- composite 163
- contexte 159
- d'accès rapide 253
- d'évitement 253
- image 162
- titre et intitulé 158
- visibilité 220

liste 199

logiciel

- de grossissement 31
- de lecture d'écran 28
- de saisie de texte 33

loi 43

loupe (pictogramme) 31

LSF (langue des signes française) 124

M

MeasureIt 268

média

- de décoration 131
- préenregistré/en direct 121
- streaming 122
- synchronisé/non synchronisé 121
- temporel/non temporel 120

MIPAW (Méthode d'intégration progressive de l'accessibilité du Web) 311

multimédia 279

- audiodescription 126
- droit d'auteur 132
- lecteur 127
- LSF 124

- sous-titrage 122
- transcription 122

N

navigation 251

- fil d'Ariane 252
- liens d'évitement/d'accès rapide 253
- liens suivant/précédent 252
- menu principal 252
- moteur de recherche 252
- nouvelles fenêtres 258
- ordre de tabulation 177
- piège au clavier 255
- plan du site 252
- système 252
- zones de ~ 253, 284

norme 58

NVDA 29

O

obligation légale 43

Orca 29, 31

P

page web

- collection de ~s 251–252
- ensemble de ~s 251–252

pixel 82

plage Braille 30

pointeur laser 33

présentation de l'information 207

propriété CSS

- @font-face 104
- background 215
- background-color 215
- display
 - none 220
- outline 224
- text-shadow 104
- transform 104
- visibility
 - hidden 220

Q

QuickJava 168

R

- rafraîchissement automatique 258
- redirection automatique 258
- référentiel 58
- région dynamique 293
- reset.css 224
- revue d'écran 28
- RGAA (référentiel général d'accessibilité pour les administrations) 46, 77, 80
 - déclaration de conformité 69
 - règles 77
 - tests unitaires 77

S

- sous-titrage 280
- structuration 286
 - de l'information 195
- surqualité 314

T

- tableau 133
 - de données 136
 - de présentation 134
 - linéarisation 134
- tabulation 177
- Tanaguru 91, 209
- technologie d'assistance 26
 - accès 184
- test utilisateur 37
- texte
 - alignement 225
 - caché 219
 - interligne 225
- titre 196
 - de la page 189
- troubles
 - de santé invalidants 25
 - DYS 22

U

- Universal Design 14

V

- vidéo 122
 - sous-titrage 122
 - transcription 122
- vieillesse 25
- VMG (Virtual Magnifying Glass) 31

W

- W3C (World Wide Web Consortium) 12, 59
 - HTML5 273
- WAI (Web Accessibility Initiative) 13, 59
- WAI-ARIA 42, 282
- WCAG (Web Content Accessibility Guidelines) 58
- WCAG 2
 - critères de succès 63
 - échecs 64
 - en un clin d'œil 66
 - niveaux de conformité 67
 - principes 61
 - règles 61
 - techniques 64
- WCAG Contrast Checker 116
- WCAG-EM (WCAG Evaluation Methodology) 309
- Web 12
 - application 281
 - riche 41, 282
 - sémantique 41, 274
- Web Developer 87, 97, 111, 136, 168, 171, 210
- widget 298

Z

- zone de navigation 284
- zoom texte seulement 216