

Notice d'accessibilité

HTML et CSS

Date	Version	Auteur	État / commentaires
06/06/2013	1.5	Atalan	

En partenariat avec :

Air Liquide – Atos – BNP Paribas – Capgemini – EDF – Generali – SFR – SNCF – Société Générale – SPIE

Et le soutien de :

Agence Entreprises & Handicap – AnySurfer (*Belgique*) – Association des Paralysés de France (APF) – Association Valentin Haüy (AVH) – CIGREF – Fondation Design For All (*Espagne*) – ESSEC – Handirect – Hanploi – Sciences Po – Télécom ParisTech

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 1/47
Réalisé par Atalan – accede @atalan.fr		Juillet 2012

Remerciements

Nous remercions tout particulièrement les 10 entreprises partenaires d'AcceDe Web et leurs équipes pour leur engagement et leur confiance :



Grâce à leur mobilisation, leur participation active aux groupes de travail, leurs relectures attentives des documents, l'utilisation test des notices sur leurs propres projets web, les entreprises partenaires ont apporté au projet AcceDe Web une expérience de terrain indispensable pour permettre la conception de notices bien adaptées aux besoins des différents intervenants d'un projet web. En permettant une diffusion libre et gratuite de ces contenus, ces entreprises contribuent à faire progresser l'accessibilité du web.

Nous remercions également les différents établissements qui nous ont apporté leur soutien et qui, dans leurs actions, sensibilisent et participent à la diffusion des notices AcceDe Web : Agence Entreprises & Handicap, AnySurfer (Belgique), Association des Paralysés de France (APF), CIGREF, ESSEC, Handirect, Hanploi, Sciences Po et Télécom ParisTech.

Nous remercions tous les membres du comité de relecture pour la qualité et la pertinence de leurs commentaires : Benjamin Ach (expert accessibilité), Vincent Aniort (APF), Jean-Baptiste Audras (Université de Grenoble), Claire Bizingre (consultante accessibilité), Victor Brito (consultant accessibilité), Anna Castilla (Provaltis), Ève Demazière (Sciences Po), Nicolas Fortin (Ministère de la Culture et de la Communication), Marc-Étienne Vargenau (Alcatel-Lucent) et Éric Vidal (Fédération des aveugles de France), ainsi que bien sûr toutes les équipes des entreprises partenaires.

Nous souhaitons enfin remercier Laurent Bracquart et Johan Ramon, Christophe Pineau et Marion Santelli qui, au sein des équipes d'Atalan, se sont impliqués sans compter pour permettre à ce projet d'être mené à bien.

Sébastien Delorme, Sylvie Goldfain,
Atalan

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 2/47
Réalisé par Atalan – accede@atalan.fr		Juillet 2012

SOMMAIRE

Introduction.....	6
Contexte et objectifs.....	6
À qui s'adresse cette notice, comment l'utiliser ?	6
Licence d'utilisation.....	7
Contact.....	7
Crédits.....	8
1. Structure générale.....	9
1.1. Régions, titres et menus de navigation.....	9
1.1.1. Identifier les principales zones de la page avec l'attribut role	9
1.1.2. Mettre en place une hiérarchie de titres logique et exhaustive avec les balises <h1> à <h6>	10
1.1.3. Structurer les menus de navigation avec des listes	11
1.2. Ordre du flux HTML.....	12
1.2.1. Écrire le code HTML en suivant la logique de l'ordre de lecture	12
1.2.2. Veiller à la cohérence de l'ordre du flux HTML d'une page à l'autre	12
2. Métadonnées	13
2.1. Titre de la page.....	13
2.1.1. Renseigner un <title> précis sur chaque page	13
2.2. Encodage	13
2.2.1. Veiller au bon codage de tous les caractères.....	13
3. Langues.....	15
3.1. Langue de la page.....	15
3.1.1. Renseigner la langue principale de la page avec l'attribut lang sur la balise <html>	15
3.2. Changement de langue	15
3.2.1. Utiliser l'attribut lang pour signaler les changements de langue	15
4. Grammaire HTML et sémantique	17
4.1. Conformité	17
4.1.1. Écrire un code HTML valide selon les règles de grammaire du DOCTYPE utilisé	17
4.2. Usage des attributs et balises HTML	17
4.2.1. Ne pas utiliser d'attributs ou de balises HTML destinés exclusivement à la mise en forme	17
4.2.2. Employer les balises HTML pour leur valeur sémantique	18
5. Images	19
5.1. Images décoratives et illustratives	19
5.1.1. Utiliser une alternative vide (alt="") sur les images décoratives et illustratives	19
5.2. Images informatives	19
5.2.1. Ne pas utiliser CSS pour afficher les images informatives.....	19
5.2.2. Renseigner l'attribut alt sur les images informatives	20
5.3. Images mappées	22
5.3.1. Renseigner l'attribut alt de chaque image mappée et de ses balises <area />.....	22
5.3.2. Intégrer les balises <map> et <area /> juste après chaque image mappée dans le flux HTML	22
6. Liens et boutons.....	23

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 3/47
Réalisé par Atalan – accede @atalan.fr		Juillet 2012

6.1. Intitulés des liens et des boutons	23
6.1.1. Rédiger des intitulés explicites pour les liens et les boutons	23
6.1.2. Utiliser en dernier recours l'attribut title pour expliciter les intitulés de liens et de boutons qui ne peuvent pas être rendus explicites.....	24
6.2. Documents en téléchargement.....	25
6.2.1. Indiquer le poids et le format de chaque document en téléchargement.....	25
6.3. Nouvelles fenêtres	25
6.3.1. Signaler l'ouverture des nouvelles fenêtres.....	25
7. Formulaires.....	27
 7.1. Intitulés et champs.....	27
7.1.1. Utiliser la balise <label> ainsi que les attributs for et id pour associer les champs à leurs intitulés	27
7.1.2. Intégrer les informations utiles à la saisie directement dans les balises <label>	27
 7.2. Regroupement et tri des informations.....	28
7.2.1. Regrouper et titrer les champs de même nature avec <fieldset> et <legend>	28
7.2.2. Ordonner les options de manière logique dans les listes déroulantes	30
 7.3. Titre de la page.....	30
7.3.1. Mettre à jour le <title> de la page quand un formulaire renvoie une erreur ou un message de confirmation	30
 7.4. CAPTCHA (systèmes anti-spam).....	31
7.4.1. Indiquer dans l'alternative des CAPTCHA graphiques où trouver la version non graphique du CAPTCHA	31
8. Listes et citations	32
 8.1. Listes	32
8.1.1. Baliser les listes non ordonnées avec et 	32
8.1.2. Baliser les listes ordonnées avec et 	32
8.1.3. Baliser les listes de définitions avec <dl>, <dt> et <dd>	32
 8.2. Citations	33
8.2.1. Baliser les blocs de citations avec <blockquote>	33
9. Tableaux	34
 9.1. Tableaux de mise en forme	34
9.1.1. Veiller à l'ordre de lecture des tableaux de mise en forme.....	34
9.1.2. Ne pas utiliser de balises ou d'attributs propres aux tableaux de données dans les tableaux de mise en forme.....	35
 9.2. Tableaux de données	36
9.2.1. Donner un titre à chaque tableau de données avec la balise <caption>	36
9.2.2. Baliser chaque cellule d'en-tête de ligne et de colonne avec <th>.....	36
9.2.3. Utiliser l'attribut scope pour associer les cellules aux entêtes des tableaux de données simples	36
9.2.4. Utiliser l'attribut headers et id pour associer les cellules aux entêtes des tableaux de données complexes	37
10. Usage des CSS	38
 10.1. Contenus masqués	38
10.1.1. Ne pas utiliser les propriétés CSS display ou visibility pour masquer un contenu qui ne peut pas être affiché par l'utilisateur	38
 10.2. Modification de l'apparence et de la taille du texte	39
10.2.1. Utiliser CSS pour mettre en forme les textes.....	39

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 4/47
Réalisé par Atalan – accede @atalan.fr		Juillet 2012

10.2.2. Utiliser uniquement des tailles relatives (em, %, small, etc.) pour les polices de caractères	39
10.2.3. Garantir la lisibilité des contenus même lorsque la taille du texte est doublée	40
10.3. Accès à l'information	40
10.3.1. Garantir la lisibilité des contenus lorsque les images ne sont pas affichées	40
10.3.2. Garantir la compréhension de l'information même lorsque CSS est désactivé	41
11. Navigation au clavier.....	42
11.1. Ordre de tabulation	42
11.1.1. Veiller à ce que l'ordre de tabulation suive la logique de l'ordre de lecture	42
11.1.2. Mettre en place des liens d'évitement.....	42
11.2. Visibilité de la prise de focus	43
11.2.1. Garantir la visibilité de la prise de focus au clavier.....	43
12. Contenus embarqués.....	45
12.1. Contenus multimédia.....	45
12.1.1. Prévoir une alternative à chaque contenu multimédia (<video>, <audio>, <object>, etc.)	45
12.1.2. Ne pas utiliser le paramètre wmode avec les valeurs transparent ou opaque	45
Annexes.....	46
Annexe 1 : recommandations additionnelles.....	46
...pour l'atteinte du niveau A (WCAG) ou Bronze (AccessiWeb)	46
...pour l'atteinte du niveau AA (WCAG) ou Argent (AccessiWeb)	46
...pour l'atteinte du niveau AAA (WCAG) ou Or (AccessiWeb).....	46
Annexe 2 : grille de correspondances WCAG/AccessiWeb/RGAA	47

AcceDe Web – www.accede-web.com	<i>Notice d'accessibilité pour la conception graphique</i>	Page 5/47
Réalisé par Atalan – accede @atalan.fr		Juillet 2012

Contexte et objectifs

Cette notice regroupe l'ensemble des spécifications à prendre en compte lors de l'intégration HTML/CSS d'un site ou d'une application web (gabarits HTML, dynamisation des gabarits, etc.) quel que soit le périphérique d'affichage ciblé (ordinateur, mobile, etc.).

Cette notice s'inscrit dans un lot de quatre notices téléchargeables sur le site www.accede-web.com :

1. Notice d'accessibilité pour la conception graphique.
2. **Notice d'accessibilité HTML/CSS (présente notice).**
3. Notice d'accessibilité interfaces riches et JavaScript.
4. Notice d'accessibilité éditoriale (modèle).



Attention

Si l'intégration HTML/CSS se fait entièrement à partir de maquettes graphiques, celles-ci doivent respecter les recommandations de la notice d'accessibilité pour la conception graphique. À l'inverse, Si l'intégration HTML/CSS est réalisée partiellement ou entièrement sans maquettes, c'est l'intégrateur qui devra veiller au respect des recommandations pour la conception graphique.

À qui s'adresse cette notice, comment l'utiliser ?

Ce document doit être transmis aux intervenants et/ou prestataires réalisant les spécifications techniques, les gabarits HTML/CSS et les différents développements techniques des interfaces. Il vient en complément aux spécifications techniques d'un projet. Les recommandations peuvent être complétées ou retirées selon les contextes d'utilisation, ce travail peut être notamment réalisé par la maîtrise d'ouvrage.

Les recommandations doivent être prises en compte en phase d'intégration HTML/CSS et, pour certaines d'entre elles, lors de la dynamisation des pages lorsque les gabarits sont affectés à un outil de gestion de contenu (CMS).

Certaines annotations viennent compléter le document, celles-ci sont nécessaires pour la compréhension de chaque recommandation :

- **Remarque** : les remarques permettent de compléter les recommandations en apportant des précisions applicables à des situations techniques précises et ponctuelles.
- **Attention** : les avertissements mettent en avant des points de vigilance particuliers qu'il est important de respecter ou des pièges à éviter pour garantir une bonne accessibilité.
- **Astuce** : les astuces ne sont pas directement liées à l'accessibilité, mais leur prise en compte permet généralement d'améliorer la qualité générale des interfaces ou de faciliter la prise en compte de l'accessibilité sur les étapes suivantes du projet. À l'inverse, notez que les recommandations de ce document, bien que dédiées à l'accessibilité, rejoignent souvent des bonnes pratiques en matière d'ergonomie, d'expérience utilisateur, de performance ou de référencement.

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 6/47
Réalisé par Atalan – accede@atalan.fr		Juillet 2012

-  **HTML5** : les encarts HTML5 apportent des éclaircissements aux critères ou des avertissements en tenant compte des évolutions proposées par la nouvelle spécification d'HTML.

Licence d'utilisation

Ce document est soumis aux termes de la licence *Creative Commons BY 3.0*¹.

Vous êtes libres :

- **de reproduire, distribuer et communiquer cette création au public,**
- **de modifier cette création,**

selon les conditions suivantes :

- **Mention de la paternité** dès lors que le document est modifié :

Vous devez mentionner clairement la mention et les logos Atalan et AcceDe Web, indiquer qu'il s'agit d'une version modifiée, et ajouter un lien vers la page où trouver l'œuvre originale : www.accede-web.com.

Vous ne devez en aucun cas citer le nom de l'auteur original d'une manière qui suggérerait qu'il vous soutient ou approuve votre utilisation de l'œuvre sans accord de sa part.

Vous ne devez en aucun cas citer les noms des entreprises partenaires (Air Liquide, Atos, BNP Paribas, Capgemini, EDF, Generali, SFR, SNCF, Société Générale et SPIE), ni ceux des soutiens (Agence Entreprises & Handicap, AnySurfer, Association des Paralysés de France (APF), CIGREF, ESSEC, Handirect, Hanploi, Sciences Po et Télécom ParisTech) sans accord de leur part.

Les marques et logos Atalan et AcceDe Web sont déposés et sont la propriété exclusive de la société Atalan. Les marques et logos des entreprises partenaires sont la propriété exclusive de Air Liquide, Atos, BNP Paribas, Capgemini, EDF, Generali, SFR, SNCF, Société Générale et SPIE.

Contact

Pour toute remarque à propos de cette notice, merci de contacter Atalan, coordinateur du projet AcceDe Web à l'adresse suivante : accede@atalan.fr.

Vous pouvez également trouver plus d'informations sur les notices méthodologiques du projet AcceDe Web sur le site www.accede-web.com ou suivre notre compte Twitter [@societe_atalan](https://twitter.com/societe_atalan).

¹ Plus d'informations sur la licence Creative Commons BY 3.0 : <http://creativecommons.org/licenses/by/3.0/fr/>.

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 7/47
Réalisé par Atalan – accede@atalan.fr		Juillet 2012

Crédits

Les icônes utilisées dans les contenus des notices AcceDe Web proviennent du set d'icônes *24x24 Free Application Icons* (<http://www.small-icons.com/packs/24x24-free-application-icons.htm>).

Les captures d'écran de l'outil de publication illustrant les bonnes pratiques sont tirées de WordPress (<http://wordpress.org>).

Les captures d'écran de contenus sont tirées des sites web listés ci-dessous, à la date du 10 octobre 2012 :

- <http://www.bnpparibas.com>
- <http://fr.atos.net>
- <http://www.generali.fr/>
- <http://www.voyages-sncf.com/>
- <http://areva.com/>
- <https://particuliers.societegenerale.fr/>
- <http://www.rff.fr/>

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 8/47
Réalisé par Atalan – accede @atalan.fr		Juillet 2012

1.1. Régions, titres et menus de navigation

1.1.1. Identifier les principales zones de la page avec l'attribut role

Les principales zones de la page doivent être identifiées à l'aide de l'attribut ARIA `role`.

Cet attribut peut prendre plusieurs valeurs pour délimiter des zones, parmi lesquelles :

- `role="banner"` pour délimiter une zone de contenu qui identifie le site, comme une bannière avec un logo, un slogan et/ou un champ de recherche.
- `role="search"` pour délimiter une zone destinée à la recherche, comme un formulaire de recherche.
- `role="navigation"` pour délimiter les menus ou les éléments de navigation principaux.
- `role="main"` pour délimiter la zone de contenu principal de la page.

L'attribut `role` peut être ajouté sur n'importe quelle balise HTML comme `<div>`, `<form>`, `<nav>`, etc.



Attention

Une page ne doit contenir qu'un seul `role="main"`.



Remarque

- Il est tout à fait possible d'imbriquer plusieurs rôles ARIA : `role="search"` dans `role="banner"`, par exemple.
- Il est recommandé de ne pas charger la page en attributs `role="navigation"` : inutile d'identifier chaque système de pagination ou chaque lien, seuls les menus principaux doivent être identifiés.



HTML5

En HTML5, une bonne pratique d'accessibilité consiste à ajouter systématiquement `role="navigation"` sur les balises `<nav>`.

```
<nav role="navigation">  
    [...]  
</nav>
```

Cette recommandation temporaire est valable jusqu'à ce que le développement de HTML5 soit finalisé, et que le langage soit bien pris en compte par les navigateurs et les aides techniques.

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 9/47
Réalisé par Atalan – accede@atalan.fr		Juillet 2012

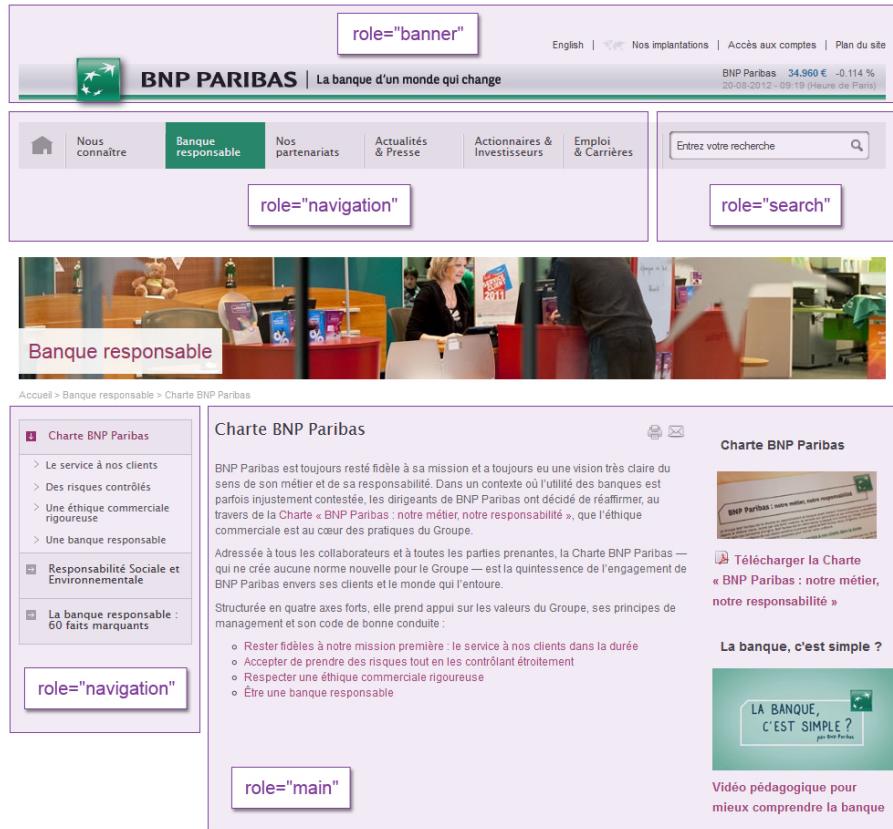


Figure 1 : Exemple de placement de rôles ARIA dans une page web.

À noter que `role="navigation"` a été ajouté sur le menu principal ainsi que sur le menu secondaire.

1.1.2. Mettre en place une hiérarchie de titres logique et exhaustive avec les balises `<h1>` à `<h6>`

Sur chaque page, pour baliser les titres, utiliser les balises de titres allant de `<h1>` jusqu'à `<h6>`, si nécessaire. La structure des titres doit être à la fois logique et exhaustive.

C'est-à-dire :

- Qu'il ne doit pas y avoir de « sauts » ni d'incohérences dans la structure des titres (passage brutal d'un `<h1>` à un `<h3>` sans `<h2>` intermédiaire, par exemple).
- Que tous les éléments qui ont valeur de titres doivent être balisés comme tels.

Astuce

Pour mettre en place une hiérarchie de titres logique et exhaustive, il faut imaginer que les titres forment la « Table des matières » de la page. Est-elle logique ? Exhaustive ?



Remarque

- Il n'est jamais gênant d'utiliser plusieurs `<h1>` dans une page si plusieurs titres de premier niveau sont présents.
- Une bonne pratique d'accessibilité consiste à ne pas ajouter de titres cachés.



HTML5

À partir de HTML5, il devient possible de mettre en place une hiérarchie de titres complète, à plusieurs niveaux, en utilisant uniquement la balise `<h1>` et les nouvelles balises destinées à créer des sections dans le document, comme `<article>` ou `<section>`. Il devient également possible de laisser des « sauts » dans la structure des titres.

Toutefois, il est préférable, pour le moment, de continuer à structurer les titres de manière logique et exhaustive, comme cela était fait avant HTML5. C'est-à-dire en utilisant les titres de `<h1>` à `<h6>`. Ceci, jusqu'à ce que le développement de HTML5 soit finalisé et que le langage soit bien pris en compte par les navigateurs et les aides techniques.

Des exemples détaillés de structures de titres pour une page sont présentés sur la version en ligne de cette recommandation : <http://wiki.accede-web.com/notices/html-css/>.

1.1.3. Structurer les menus de navigation avec des listes

Utiliser des listes non ordonnées `` et `` pour baliser les menus de navigation.

Dans le cas d'un menu à plusieurs niveaux, veiller à la bonne imbrication des listes :

```
<ul id="menu-principal" role="navigation">
    <li><a href="page-1.html">Premier item du menu</a></li>
    <li>
        <a href="page-2.html">Deuxième item du menu</a>
        <ul>
            <li><a href="page-3.html">Premier item du sous-menu</a></li>
            <li><a href="page-4.html">Deuxième item du sous-menu</a></li>
        </ul>
    </li>
    <li><a href="page-5.html">Troisième item du menu</a></li>
</ul>
```



HTML5

En HTML5, englober en plus les menus de navigation *principaux* avec une balise `<nav>`, de façon à obtenir :

```
<nav id="menu-principal" role="navigation">
    <ul>
        [...]
    </ul>
</nav>
```

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 11/47
Réalisé par Atalan – accede@atalan.fr		Juillet 2012

1.2. Ordre du flux HTML

1.2.1. Écrire le code HTML en suivant la logique de l'ordre de lecture

L'ordre d'écriture des balises dans le flux HTML doit suivre la logique de l'ordre de lecture de la page.

C'est-à-dire que lorsqu'une balise précède immédiatement une autre balise lors du parcours visuel de la page, la première doit également précéder immédiatement la seconde dans le flux HTML.



Astuce

Pour tester ce critère, il suffit de désactiver les CSS est de s'assurer que le résultat obtenu correspond bien à l'ordre de lecture de la page, lorsque les CSS sont activées.



Attention

Si des contenus sont masqués par défaut, veiller à leur bon positionnement dans le flux HTML lorsque les styles sont désactivés.

1.2.2. Veiller à la cohérence de l'ordre du flux HTML d'une page à l'autre

L'ordre d'intégration des principaux blocs dans le flux HTML doit rester cohérent d'une page à l'autre du site.

Par exemple, si le menu secondaire est placé après le contenu principal, il est important de conserver cet ordre sur l'ensemble des pages du site.

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 12/47
Réalisé par Atalan – accede@atalan.fr		Juillet 2012

2.1. Titre de la page

2.1.1. Renseigner un <title> précis sur chaque page

Sur chaque page, la balise <title> doit être renseignée avec précision.

Elle doit au minimum annoncer le nom de la page courante ainsi que le nom du site.



Remarque

Le nom de la page courante doit apparaître en premier dans la balise <title>. Il est par exemple possible d'opter pour <title>[Nom de la page courante] | [Nom du site]</title>.



Attention

Parfois, certaines pages sont rechargées avec un contenu modifié, suite à une action de l'utilisateur. C'est le cas :

- Lors de l'utilisation de filtres, comme des nuages de mots-clés.
- Lors de l'utilisation de la pagination.
- Lorsqu'une expression est recherchée via le formulaire de recherche.
- Etc.

Il faut alors veiller à mettre à jour le titre de la page en conséquence, par exemple :

```
<title>Résultats de votre recherche sur "[Expression recherchée]" (page 3/7)  
| [Nom du site]</title>
```



Astuce

Une bonne pratique d'accessibilité consiste à veiller à la cohérence de l'ordre des contenus de la balise <title> sur l'ensemble des pages du site.

2.2. Encodage

2.2.1. Veiller au bon codage de tous les caractères

Afin de garantir la bonne restitution des contenus textuels à l'utilisateur, une déclaration de codage des caractères doit être effectuée sur chaque page, par exemple à l'aide de la balise <meta> suivante :

```
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
  [...]  
</head>
```

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 13/47
Réalisé par Atalan – accede@atalan.fr		Juillet 2012

Cette balise `<meta>` doit être insérée le plus haut possible dans la page HTML, avant tout contenu affiché à l'écran, et, de préférence, immédiatement après l'ouverture de la balise `<head>`.

 **Attention**

Penser également à vérifier le bon codage des caractères qui ne sont pas spontanément affichés à l'écran, notamment :

- Au contenu des attributs `title`.
- Aux textes cachés par défaut.
- Au contenu des attributs `alt` des images.
- Etc.

 **HTML5**

HTML5 simplifie l'écriture de la balise `<meta>` destinée à la déclaration de codage. Ainsi, si vous optez pour ce type de document, il suffit alors d'utiliser par exemple : `<meta charset="UTF-8">`.

3.1. Langue de la page

3.1.1. Renseigner la langue principale de la page avec l'attribut lang sur la balise <html>

Afin de garantir la bonne restitution des contenus textuels à l'utilisateur, une déclaration de langue principale doit être effectuée sur chaque page. Utiliser pour cela l'attribut `lang` sur la balise `<html>`.

Par exemple, pour une page en français : `<html lang="fr">`.



Astuce

Pour renseigner l'attribut `lang`, un code de langue ISO 639-1 (sur deux lettres) ou, s'il n'est pas disponible, ISO 639-2 (sur trois lettres) doit être utilisé.

Les principaux codes utilisés sont les suivants :

- `fr` pour le français.
- `en` pour l'anglais.
- `es` pour l'espagnol.
- `de` pour l'allemand.
- `it` pour l'italien.

La liste exhaustive des codes est maintenue à jour à cette adresse :
http://www.loc.gov/standards/iso639-2/php/French_list.php.

3.2. Changement de langue

3.2.1. Utiliser l'attribut lang pour signaler les changements de langue

Si des contenus sont proposés dans une langue différente de la langue principale, alors ils doivent être signalés avec l'attribut `lang`.

Par exemple, dans le cas d'une page en français :

```
<ul>
    <li>Version française</li>
    <li><a href="http://website.co.uk" lang="en">English version</a></li>
</ul>
```



Astuce

Si aucune balise n'encadre directement le contenu en langue étrangère, alors utiliser la balise `` ou `<div>` et renseigner son attribut `lang`.

AcceDe Web – www.accede-web.com	Notice d'accessibilité pour la conception graphique	Page 15/47
Réalisé par Atalan – accede@atalan.fr		Juillet 2012



Remarque

Signaler le changement de langue n'est pas demandé pour :

- Les noms propres.
- Les mots d'origine étrangère à la base, mais intégrés dans le dictionnaire de la langue principale.
- Tous les mots d'origine étrangère, mais qui se prononcent et se comprennent correctement avec l'accent de la langue principale (« podcast », par exemple, si la langue principale est le français).

4.1. Conformité

4.1.1. Écrire un code HTML valide selon les règles de grammaire du DOCTYPE utilisé

Sur chaque page, un DOCTYPE doit être utilisé et le code HTML doit être valide selon les règles de grammaire de ce dernier (le choix du DOCTYPE est libre).

Il est particulièrement important de veiller à :

- La correcte imbrication des balises.
- La correcte ouverture et fermeture des balises.
- L'absence d'attributs dupliqués sur une même balise.
- L'unicité des valeurs de l'attribut id au sein d'une même page.



Remarque

Seules les erreurs de validation sont à corriger. Les avertissements renvoyés par le validateur ne sont pas à prendre en compte dans le cadre de ce critère car sans impact sur l'accessibilité.



Astuces

- La liste des DOCTYPE officiels est disponible à cette adresse : <http://www.w3.org/QA/2002/04/valid-dtd-list.html>.
- Utiliser le validateur du W3C pour tester la validité de votre code HTML : <http://validator.w3.org/>.



Attention

La mise en place d'attributs ARIA dans le code HTML est un vrai bénéfice pour l'accessibilité mais invalidera le code source si la page n'est pas servie en HTML5. Le cas échéant, les erreurs remontées suite à la mise en place de ces attributs ne sont pas à corriger.

4.2. Usage des attributs et balises HTML

4.2.1. Ne pas utiliser d'attributs ou de balises HTML destinés exclusivement à la mise en forme

Les balises et attributs HTML destinés exclusivement à la mise en forme ne doivent pas être utilisés :

- C'est le cas des attributs : align, alink, background, basefont, bgcolor, border, color, link, text, vlink, height et width.
- Ainsi que les balises : <basefont>, <blink>, <center>, , <marquee>, <s>, <strike>, <tt> et <u>.

Utiliser CSS pour obtenir le rendu visuel souhaité.

4.2.2. Employer les balises HTML pour leur valeur sémantique

Les balises HTML doivent être utilisées pour leur valeur sémantique, et non pour leur rendu visuel.

Par exemple :

Balises	Mauvais usage	Bon usage
<blockquote>	Indenter le texte	Baliser un bloc de citation
 et 	Obtenir une puce à l'écran	Baliser des listes
<fieldset>	Obtenir une bordure	Regrouper des éléments de formulaires de même nature
<h1> à <h6>	Obtenir des textes plus ou moins grands	Baliser des titres et sous-titres

5. Images

5.1. Images décoratives et illustratives

5.1.1. Utiliser une alternative vide (`alt=""`) sur les images décoratives et illustratives

Lorsqu'une image décorative ou illustrative est intégrée dans le code HTML, l'attribut `alt` doit être ajouté sur la balise `` et laissé vide (sans aucun espace entre les guillemets de `alt=""`).

```

```



Astuce

Afin de savoir à coup sûr si une image est décorative/illustrative ou informative, se demander si l'absence de celle-ci altérerait la compréhension du contenu. Si oui : renseigner le texte de remplacement ; si non : laisser le texte de remplacement vide.



Remarque

Autant que possible, les images décoratives ne doivent pas être intégrées dans le code HTML, mais chargées dans la page via CSS.



Figure 2 : dans cet exemple, les images décoratives et illustratives sont pointées. Si elles sont intégrées dans le code HTML et non cliquables elles doivent alors disposer d'une alternative vide (`alt=""`).

5.2. Images informatives

5.2.1. Ne pas utiliser CSS pour afficher les images informatives

Les CSS ne doivent **jamais** être utilisées pour afficher les images porteuses d'informations.

AcceDe Web – www.accede-web.com	Notice d'accessibilité HTML et CSS	Page 19/47
Réalisé par Atalan – accede@atalan.fr		Octobre 2012

En d'autres termes, chaque fois qu'une image apporte de l'information, elle doit être intégrée en dur dans le code HTML, la plupart du temps avec la balise ``.



Remarque

L'usage de *sprites* (images chargées en arrière-plan via CSS) n'est donc pas autorisé pour les images porteuses d'informations.



Astuce

Sont considérées comme informatives toutes les images qui entraîneraient une perte de contenus ou de fonctionnalités, si elles n'étaient pas disponibles :

- Logos.
- Images-textes.
- Liens et boutons-images.
- Etc.



Attention

En particulier, la technique qui consiste à sortir un texte de l'écran pour le remplacer par une image d'arrière-plan ne doit jamais être utilisée.

Lorsque les styles ne sont pas chargés en totalité, ou qu'ils sont modifiés par l'utilisateur, l'information peut être perdue.

```
<a href="/" id="logo">[Le texte de mon logo]</a>

a#logo
{
display: block;
background: url(images/logo.png);
text-indent: -999999px;
width: 300px;
height: 100px;
}
```

Cette technique est à remplacer par l'utilisation de HTML pur. Par la suite, c'est le JavaScript qui doit être utilisé pour modifier dynamiquement l'image au survol si besoin.

```
<a href="/" id="logo"></a>
```

Si l'image n'est pas chargée, alors le texte de remplacement sera affiché à la place et l'information ne sera pas perdue.

5.2.2. Renseigner l'attribut alt sur les images informatives

Lorsqu'une image informative est intégrée dans le code HTML :

- Un attribut alt doit être ajouté sur la balise ``.

- Cet attribut `alt` doit être renseigné avec une information égale ou équivalente à celle véhiculée par l'image.



Attention

Il est fortement déconseillé de rédiger le texte de remplacement d'une image en commençant par `alt="Image [...]"`. Cette information est implicite et pourra être restituée aux utilisateurs d'aides techniques.



Remarque : les images véhiculant des informations riches

Parfois, l'image véhicule des informations riches qu'il n'est pas possible de retranscrire sous forme de texte brut, via le simple attribut `alt` (schémas, graphiques complexes, etc.).

Dans cette situation, une bonne pratique d'accessibilité consiste à :

- Renseigner un **court texte de remplacement** dans l'attribut `alt` qui décrit globalement la fonction de l'image.
- Prévoir une zone de contenu dans laquelle proposer une **description détaillée de l'image**. De préférence, soit directement sous l'image, soit par l'intermédiaire d'un lien vers une autre page. Dans le cas de graphiques ou schémas complexes, la description détaillée la plus adaptée est souvent un tableau de données ou une liste.
- Indiquer dans le texte de remplacement de l'image le moyen d'accéder à la description plus détaillée.

Par exemple, dans le cas d'une carte du monde qui expose les différents pays dans lesquels une marque est présente :

```

```

```
<a href="/zones-d-implantations/">Voir la liste des pays dans lesquels la marque Spartacus est présente</a>
```

À noter que l'attribut `longdesc` est parfois recommandé pour fournir un lien vers la description détaillée d'une image. Cependant, l'utilisation de cet attribut ne garantit pas l'accès à l'information pour tous, contrairement à l'ajout d'un lien en dur dans la page. L'emploi de la technique proposée ci-dessus est donc à privilégier.



Figure 3 : dans cet exemple, les images informatives sont pointées. Elles doivent être intégrées dans le code HTML et disposer d'une alternative reprenant les textes inclus dans les images tel que et .

5.3. Images mappées

5.3.1. Renseigner l'attribut alt de chaque image mappée et de ses balises <area />

Lorsqu'une image mappée est intégrée dans le code HTML :

- Un attribut alt doit être rajouté sur la balise `` ainsi que sur chaque balise `<area />`.
- L'attribut alt de la balise `` doit annoncer la fonction de l'image mappée.
- L'attribut alt de chaque balise `<area />` doit exprimer la fonction du lien.

Par exemple, dans le cas d'une carte de France où chaque département est cliquable et mène l'utilisateur vers une fiche associée au département :

```

<map name="carte-france">
  <area shape="poly" coords="[...]" href="ain.html" alt="Ain (01)" />
  <area shape="poly" coords="[...]" href="aisne.html" alt="Aisne (02)" />
  <area shape="poly" coords="[...]" href="allier.html" alt="Allier (03)" />
  [...]
</map>
```

5.3.2. Intégrer les balises <map> et <area /> juste après chaque image mappée dans le flux HTML

Chaque fois qu'une image mappée est utilisée, les balises `<map>` et `<area />` doivent immédiatement suivre la balise `` associée dans l'ordre du flux HTML.

6.1. Intitulés des liens et des boutons

6.1.1. Rédiger des intitulés explicites pour les liens et les boutons

Autant que possible, l'intitulé des liens et des boutons doit être explicite. C'est-à-dire que la fonction du lien ou du bouton doit être parfaitement compréhensible à la lecture du seul intitulé.

Cet intitulé correspond au texte contenu :

- Entre les balises `<a>` et ``.
- Entre les balises `<button>` et `</button>`.
- Dans l'attribut `alt` dans le cas d'un `<input type="image" />`.
- Dans l'attribut `value` dans le cas des `<input type="button" />` ou `<input type="submit" />`.

Ainsi, plutôt que des boutons « OK » ou des liens « Retour », privilégier :

```
<input type="submit" value="S'inscrire" />  
  
<a href="panier.html">Retour à l'étape 1 : "Votre panier"</a>
```



Astuce

Pour rédiger des intitulés explicites, il faut imaginer que les intitulés sont extraits de la page et lus en dehors de tout contexte. Restent-ils compréhensibles ? Permettent-ils de se faire une idée du contenu de la page de destination ou de l'action à effectuer ?



Remarque

Dans le cas où du texte et de l'image sont présents simultanément dans l'intitulé du lien ou du bouton, le texte de remplacement de l'image est à considérer comme faisant partie intégrante de l'intitulé.

Ainsi, l'intitulé du lien suivant sera considéré comme explicite :

```
<a href="/derniers-changements/">  
      
    Veillez à prendre en considération les derniers changements apportés à  
    Canelis avant toute utilisation.  
</a>
```



Attention

Les liens et les boutons ne doivent jamais être vides et doivent donc toujours posséder un intitulé.

AcceDe Web – www.accede-web.com	Notice d'accessibilité HTML et CSS	Page 23/47
Réalisé par Atalan – accede@atalan.fr		Octobre 2012

6.1.2. Utiliser en dernier recours l'attribut title pour expliciter les intitulés de liens et de boutons qui ne peuvent pas être rendus explicites

Certains intitulés de liens ou de boutons ne peuvent pas être rendus explicites (manque d'espace disponible à l'écran, choix éditorial, etc.). C'est souvent le cas par exemple pour les liens du type « Lire la suite » ou « En savoir plus ».

Dans ces situations, c'est l'attribut `title` qui doit alors être utilisé pour préciser la fonction du lien ou du bouton.

La valeur de l'attribut `title` doit successivement :

- Reprendre une information identique ou équivalente à celle véhiculée par l'intitulé.
- Compléter l'information fournie par l'intitulé.

Par exemple, dans le cas d'un lien « Juillet 2012 » :

```
<a href="archives/2012/07/" title="Juillet 2012 (lire les articles du mois)">Juillet  
2012</a>
```

Attention

L'utilisation de l'attribut `title` pour expliciter un intitulé sera toujours moins accessible que l'optimisation directe de l'intitulé. Le contenu de l'attribut `title` ne s'affiche pas par défaut lors de la navigation au clavier ou sur un écran tactile, par exemple.

Cet attribut ne doit donc être utilisé qu'en dernier recours.

Astuce

Dans le cas de liens du type « Lire la suite » ou « En savoir plus », une bonne pratique d'accessibilité consiste à reléguer cette information en dernier dans le contenu de l'attribut `title` afin de faciliter l'accès à l'information importante :

```
<a href="entree-en-bourse.html" title="Entrée en bourse de la société  
Boursicota (lire la suite)">Lire la suite</a>
```

Remarque

L'attribut `title` doit également être utilisé pour distinguer les liens ou boutons dont les intitulés pourraient être considérés comme explicites, mais qui pointent vers des pages ou déclenchent des actions différentes.

Par exemple, dans le cas de deux boutons « Rechercher » affichés sur la même page, et qui seraient à distinguer :

```
<input type="submit" value="Rechercher" title="Rechercher une actualité" />  
[...]  
<input type="submit" value="Rechercher" title="Rechercher une personne dans  
l'annuaire" />
```

6.2. Documents en téléchargement

6.2.1. Indiquer le poids et le format de chaque document en téléchargement

Chaque fois qu'un lien ou un bouton pointe sur un document à télécharger, les informations suivantes doivent être intégrées **dans** l'intitulé :

- Le nom du document.
- Le format du document.
- Le poids du document.

Par exemple :

```
<a href="rapport-annuel-2011.pdf">Télécharger le rapport annuel 2011 (PDF, 23 Mo)</a>
```



Astuce

Si des pictogrammes sont utilisés pour indiquer visuellement le format des documents en téléchargement, penser à bien intégrer l'image dans l'intitulé, en dur dans le code HTML, et à bien renseigner le texte de remplacement de cette dernière.

Par exemple :

```
<a href="rapport-annuel-2011.pdf">  
    Télécharger le rapport annuel 2011 (23 Mo)  
      
</a>
```



Remarque

Parfois, l'intitulé ne peut être modifié pour intégrer ces informations. Dans cette situation, et en dernier recours, utiliser l'attribut `title` sur le lien ou le bouton pour reprendre et compléter l'intitulé.

6.3. Nouvelles fenêtres

6.3.1. Signaler l'ouverture des nouvelles fenêtres

Chaque fois qu'un lien ou un bouton déclenche l'ouverture forcée d'une nouvelle fenêtre dans le navigateur, l'utilisateur doit être prévenu :

- Soit par l'ajout d'une mention du type « (nouvelle fenêtre) », directement dans l'intitulé.
- Soit par l'ajout d'une image dans l'intitulé, avec texte de remplacement renseigné.
- Soit par l'ajout d'une mention du type « (nouvelle fenêtre) », dans l'attribut `title`.

Par exemple :

```
<a href="cgv.html" target="_blank">Conditions Générales de Vente (nouvelle  
fenêtre)</a>
```

```
<a href="cgv.html" target="_blank">  
    Conditions Générales de Vente
```

```
  
</a>  
  
<a href="cgv.html" target="_blank" title="Conditions Générales de Vente (nouvelle  
fenêtre)">Conditions Générales de Vente</a>
```

7.1. Intitulés et champs

7.1.1. Utiliser la balise `<label>` ainsi que les attributs `for` et `id` pour associer les champs à leurs intitulés

Chaque champ de formulaire doit être associé à un intitulé.

Pour cela, procéder de la sorte :

1. Utiliser `<label>` pour baliser chaque intitulé.
2. Ajouter un attribut `for` sur chaque balise `<label>` ainsi qu'un attribut `id` sur chaque champ.
3. Renseigner avec une valeur identique les attributs `id` et `for` de chaque couple intitulé/champ.

```
<label for="nom">Votre nom</label>
<input type="text" id="nom" name="nom" />

[...]

<label for="annee-naissance">Votre année de naissance</label>
<select id="annee-naissance" name="annee-naissance">
  <option value="2012">2012</option>
  <option value="2011">2011</option>
  <option value="2010">2010</option>
  <option value="2009">2009</option>
  [...]
</select>
```



Remarque

Parfois, certains champs n'ont pas d'intitulé visible. Dans cette situation, utiliser l'attribut `title` pour associer tout de même un intitulé au champ.

```
<input type="text" title="Votre recherche" name="recherche" />
```

À noter que l'attribut `title` ne doit jamais être utilisé si le couple d'attributs `for` et `id` est déjà en place pour associer un intitulé à un champ.



Attention

Il est important de prévoir des intitulés identiques pour les champs dont la fonction est identique.

Par exemple, si plusieurs formulaires d'identification sont présents dans le site, ne pas utiliser l'intitulé « Identifiant » pour l'un et « Login » pour l'autre.

7.1.2. Intégrer les informations utiles à la saisie directement dans les balises `<label>`

Les informations utiles à la saisie doivent être intégrées directement dans les balises `<label>`.

C'est notamment le cas :

- Des mentions du type « Champ obligatoire ».
- Des indications qui permettent de connaître le format de saisie attendu, du type « JJ/MM/AAAA » pour un format de date.
- Des indications qui permettent de connaître le type et le poids maximal autorisé pour l'envoi de fichiers.
- Etc.

```
<p>Les champs marqués d'un astérisque (*) sont obligatoires.</p>
```

```
<label for="nom">Votre nom *</label>
<input type="text" id="nom" name="nom" />

[...]

<label for="numero-client">
    Votre numéro de client
    <input type="text" id="numero-client" name="numero-client" />
    <em>Par exemple : 76432-BT-VZ</em>
</label>
```



Remarque

Parfois, les messages d'erreurs sont affichés au niveau de chaque champ. Dans ce cas, ils doivent également être intégrés dans les `<label>` associés.

```
<p>Les champs marqués d'un astérisque (*) sont obligatoires.</p>
```

[...]

```
<label for="email">
    Votre adresse email *
    <input type="text" id="email" name="email" />
    <span class="erreur">Veuillez renseigner votre adresse email (format
attendu : exemple@domaine.fr).</span>
</label>
```

7.2. Regroupement et tri des informations

7.2.1. Regrouper et titrer les champs de même nature avec `<fieldset>` et `<legend>`

Lorsqu'un formulaire propose plusieurs groupes de champs, dont certains ont des intitulés identiques, utiliser les balises `<fieldset>` et `<legend>`.

Par exemple :

```
<fieldset>
    <legend>Participant 1</legend>

    <label for="prenom-1">Prénom</label>
    <input type="text" id="prenom-1" name="prenom-1" />

    <label for="nom-1">Nom</label>
    <input type="text" id="nom-1" name="nom-1" />

    [...]
</fieldset>

<fieldset>
```

```

<legend>Participant 2</legend>

<label for="prenom-2">Prénom</label>
<input type="text" id="prenom-2" name="prenom-2" />

<label for="nom-2">Nom</label>
<input type="text" id="nom-2" name="nom-2" />

[...]

</fieldset>

```

Attention

Les balises `<fieldset>` et `<legend>` ne sont à utiliser que lorsque plusieurs groupes de champs disposent d'intitulés identiques. Par exemple :

- Une série de questions sur une même page avec comme réponses possibles "oui" ou "non".
- Une liste de participants à un évènement avec, à chaque fois, "nom" et "prénom".

Dans toutes les autres situations :

- Ne pas utiliser les balises `<fieldset>` et `<legend>`.
- Utiliser les balises `<h1>` à `<h6>` pour titrer les blocs.



Remarque

Une bonne pratique d'accessibilité consiste à utiliser également les balises `<fieldset>` et `<legend>` lors de l'intégration de listes de boutons radio ou de cases à cocher dans la page.

Par exemple :

```

<fieldset>
    <legend>Civilité</legend>
    <ul>
        <li>
            <label for="madame">
                <input id="madame" type="radio" name="civilite"
value="Mme" />
                Madame
            </label>
        </li>
        <li>
            <label for="monsieur">
                <input id="monsieur" type="radio" name="civilite"
value="M." />
                Monsieur
            </label>
        </li>
    </ul>
</fieldset>

```

Des exemples détaillés d'utilisation de `<fieldset>` et `<legend>` pour un formulaire sont présentés sur la version en ligne de cette recommandation : <http://wiki.accede-web.com/notices/html-css/>.

7.2.2. Ordonner les options de manière logique dans les listes déroulantes

Lorsque des listes déroulantes sont utilisées, les options qu'elles contiennent doivent être ordonnées de manière logique.

L'ordre logique dépend du contexte, mais il peut s'agir notamment :

- D'un ordre alphabétique (une liste de langues, par exemple).
- D'un ordre numérique (une liste de départements, par exemple).
- D'un ordre pratique (« France » en premier dans un formulaire d'inscription à un service français).



Astuce

Une bonne pratique d'accessibilité consiste à ne pas préfixer le contenu de la balise `<option>` avec des caractères décoratifs (tirets, étoiles, espaces, etc.).

Ceci afin de permettre un accès direct à une valeur ou à un groupe de valeurs souhaité, par simple pression d'une touche au clavier (pression sur la touche « I » pour atteindre le pays « Italie », par exemple).

Ainsi :

```
<select>
  <option>--Allemagne</option>
  <option>--Belgique</option>
  <option>--Espagne</option>
  <option>--France</option>
  <option>--Italie</option>
</select>
```

Sera remplacé avantageusement par :

```
<select>
  <option>Allemagne</option>
  <option>Belgique</option>
  <option>Espagne</option>
  <option>France</option>
  <option>Italie</option>
</select>
```

7.3. Titre de la page

7.3.1. Mettre à jour le `<title>` de la page quand un formulaire renvoie une erreur ou un message de confirmation

Chaque fois qu'un formulaire renvoie une erreur ou un message de confirmation, la balise `<title>` de la page doit être mise à jour.

Par exemple, dans le cas d'une réussite :

```
<title>Confirmation - Formulaire de contact | [Nom du site]</title>
```

Et, dans le cas d'une erreur :

AcceDe Web – www.accede-web.com	Notice d'accessibilité HTML et CSS	Page 30/47
Réalisé par Atalan – accede@atalan.fr		Octobre 2012



Remarque

Dans certaines situations, il n'est pas nécessaire d'ajouter une mention car le titre de la page affichée après la soumission rend le résultat de l'action évident. Par exemple :

- Formulaire de connexion qui envoie vers une page "Profil utilisateur".
- Bouton "Passer à l'étape suivante" qui envoie à l'étape suivante dans un formulaire à étapes multiples.
- Formulaire de contact qui envoie une page de prévisualisation.
- Etc.

7.4. CAPTCHA (systèmes anti-spam)

7.4.1. Indiquer dans l'alternative des CAPTCHA graphiques où trouver la version non graphique du CAPTCHA

Chaque fois qu'un CAPTCHA graphique est utilisé, indiquer dans son alternative textuelle où se trouve la version non graphique du CAPTCHA.

Veuillez taper les caractères que vous voyez sur l'image

Écouter le code
Essayer une autre image

Veuillez inscrire les lettres lues dans le champ suivant*

Souvent, il s'agit d'une version audio, opter alors par exemple pour :

```

```

8.1. Listes

8.1.1. Baliser les listes non ordonnées avec et

Utiliser les balises `` et `` pour baliser les listes d'éléments pour lesquelles l'ordre n'a pas d'importance (menus, onglets, boutons de partage, plan du site, etc.).

Le cas échéant, veiller à la bonne imbrication des listes :

```
<ul>
    <li>Premier item de premier niveau</li>
    <li>
        Deuxième item de premier niveau
        <ul>
            <li>Premier item de second niveau</li>
            <li>Deuxième item de second niveau</li>
        </ul>
    </li>
    <li>Troisième item de premier niveau</li>
</ul>
```

8.1.2. Baliser les listes ordonnées avec et

Utiliser les balises `` et `` pour baliser les listes d'éléments pour lesquelles l'ordre a de l'importance (liste d'étapes dans un processus, classement quelconque, etc.). C'est-à-dire lorsque l'information ne serait plus comprise si les éléments étaient placés dans un ordre différent.

8.1.3. Baliser les listes de définitions avec <dl>, <dt> et <dd>

Utiliser les balises `<dl>`, `<dt>` et `<dd>` pour baliser les listes de définitions (dans un glossaire par exemple).

Il est possible d'associer plusieurs définitions à un seul terme en procédant de la manière suivante :

```
<dl>
    <dt>Terme 1</dt>
    <dd>Définition du terme 1.</dd>
    <dt>Terme 2</dt>
    <dt>1. Première définition du terme 2.</dt>
    <dt>2. Deuxième définition du terme 2.</dt>
</dl>
```

HTML5

La spécification HTML5 étend le rôle de la liste de définitions classique à celui plus large d'une liste de descriptions : il est désormais possible d'utiliser `<dl>`, `<dt>` et `<dd>` pour baliser n'importe quels groupes de clés/valeurs (caractéristiques dans une fiche produit, informations de publication d'un article, etc.).

```
<h2>Conférence sur le développement durable</h2>
<dl>
  <dt>Lieu</dt>
  <dd>Paris</dd>
  <dt>Date</dt>
  <dd>Samedi 29 septembre 2012</dd>
  <dt>Heure</dt>
  <dd>À partir de 10 heures</dd>
</dl>
```

8.2. Citations

8.2.1. Baliser les blocs de citations avec `<blockquote>`

Chaque fois qu'un bloc de citation est à intégrer dans la page HTML, entourer ce dernier d'une balise `<blockquote>`.



Remarque

Toute citation isolée, qui n'est pas incorporée directement en ligne dans le flux du texte environnant, est considérée comme un bloc de citation.

9. Tableaux

9.1. Tableaux de mise en forme

9.1.1. Veiller à l'ordre de lecture des tableaux de mise en forme

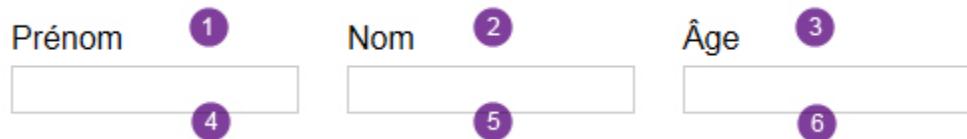
Lorsqu'une personne utilise un lecteur d'écran pour accéder à un tableau de mise en forme, son contenu est restitué de manière linéaire. C'est-à-dire que le contenu du tableau est lu cellule après cellule, de gauche à droite et ligne après ligne.

Il faut donc veiller à ce que l'ordre de lecture reste cohérent dans chaque tableau de mise en forme.

Par exemple, si le code source qui suit est utilisé, l'ordre de lecture est :

1. Prénom.
2. Nom.
3. Âge.
4. Champ « Prénom ».
5. Champ « Nom ».
6. Champ « Âge ».

```
<table>
  <tr>
    <td>
      <label for="prenom">Prénom</label>
    </td>
    <td>
      <label for="nom">Nom</label>
    </td>
    <td>
      <label for="age">Âge</label>
    </td>
  </tr>
  <tr>
    <td>
      <input type="text" name="prenom" id="prenom" />
    </td>
    <td>
      <input type="text" name="nom" id="nom" />
    </td>
    <td>
      <input type="text" name="age" id="age" />
    </td>
  </tr>
</table>
```



Pour obtenir un ordre de lecture cohérent tout en conservant le tableau de mise en forme, opter pour :

```
<table>
  <tr>
    <td>
```

```

<label for="prenom">Prénom</label>
<input type="text" name="prenom" id="prenom" />
</td>
<td>
    <label for="nom">Nom</label>
    <input type="text" name="nom" id="nom" />
</td>
<td>
    <label for="age">Âge</label>
    <input type="text" name="age" id="age" />
</td>
</tr>
</table>

```

Prénom	1	Nom	3	Âge	5
<input type="text"/>	2	<input type="text"/>	4	<input type="text"/>	6



Afin de faciliter la lecture par les aides techniques et de garantir l'ordre de lecture, il est fortement recommandé de limiter l'imbrication des tableaux de mise en forme.

9.1.2. Ne pas utiliser de balises ou d'attributs propres aux tableaux de données dans les tableaux de mise en forme

Un tableau de mise en forme ne doit pas posséder de balises ou d'attributs propres aux tableaux de données.

C'est-à-dire :

- Que les balises `<caption>`, `<th>`, `<thead>` et `<tfoot>` ne doivent pas être utilisées dans les tableaux de mise en forme.
- Que les attributs `scope`, `headers`, `axis`, `colgroup` et `summary` ne doivent pas être utilisés dans les tableaux de mise en forme.



Astuce

Il est intéressant de signaler les tableaux de mise en forme en utilisant l'attribut `role` de la façon suivante : `<table role="presentation">`.

9.2. Tableaux de données

9.2.1. Donner un titre à chaque tableau de données avec la balise <caption>

Chaque fois qu'un tableau de données est intégré dans la page HTML, celui-ci doit posséder un titre clair et concis. Ce titre doit être balisé avec `<caption>` et doit introduire le contenu du tableau.

```
<table>
  <caption>Températures moyennes mensuelles des 3 plus grandes villes de
France en 2012.</caption>
  [...]
</table>
```

9.2.2. Baliser chaque cellule d'entête de ligne et de colonne avec <th>

Dans chaque tableau de données, baliser chaque cellule d'entête de ligne et de colonne avec la balise `<th>`. C'est-à-dire que chaque fois qu'une cellule est nécessaire à la compréhension des données proposées dans le tableau, celle-ci doit être balisée avec `<th>`.

9.2.3. Utiliser l'attribut scope pour associer les cellules aux entêtes des tableaux de données simples

Un tableau de données simple est un tableau dans lequel les cellules d'entêtes s'appliquent systématiquement à la totalité des cellules d'une ligne ou d'une colonne.

Pour associer les entêtes à leurs données dans ce type de tableaux, utiliser l'attribut `scope` sur les balises `<th>`. La valeur de cet attribut changera selon que la cellule d'entête concerne :

- La totalité d'une colonne : `scope="col"`.
- La totalité d'une ligne : `scope="row"`.

```
<table>
  <caption>Températures moyennes mensuelles des 3 plus grandes villes de
France en 2012.</caption>
  <tr>
    <td>&nbsp;</td>
    <th scope="col">Paris</th>
    <th scope="col">Marseille</th>
    <th scope="col">Lyon</th>
  </tr>
  <tr>
    <th scope="row">Juin</th>
    <td>22°C</td>
    <td>28°C</td>
    <td>26°C</td>
  </tr>
  <tr>
    <th scope="row">Juillet</th>
    <td>24°C</td>
    <td>30°C</td>
    <td>28°C</td>
  </tr>
</table>
```

9.2.4. Utiliser l'attribut headers et id pour associer les cellules aux entêtes des tableaux de données complexes

Un tableau de données complexe est un tableau dans lequel les cellules d'entêtes ne s'appliquent pas systématiquement à la totalité des cellules de données d'une ligne ou d'une colonne.

Pour associer les entêtes aux données dans ce type de tableaux, utiliser les attributs `id` (identifiants) sur les cellules `<th>` et `headers` sur les cellules `<td>`.

Il suffit ensuite de renseigner l'attribut `headers` avec les identifiants des cellules d'entêtes associées. Si plusieurs entêtes sont associés à une cellule de données, les identifiants doivent être séparés par des espaces dans `headers`.

```
<table>
  <caption>Comparatif du chiffre d'affaires des entreprises Dupond et Dupont en France et dans le monde en 2011</caption>
  <tr>
    <th id="entete-un">En millions d'euros</th>
    <th id="entete-deux">En France</th>
    <th id="entete-trois">Dans le monde</th>
  </tr>
  <tr>
    <th id="entete-quatre">Dupond</th>
    <td headers="entete-quatre entete-deux entete-un">50,7</td>
    <td headers="entete-quatre entete-trois entete-un">139,3</td>
  </tr>
  <tr>
    <th id="entete-cinq">Dupont</th>
    <td headers="entete-cinq entete-deux entete-un">27,1</td>
    <td headers="entete-cinq entete-trois entete-un">476,0</td>
  </tr>
</table>
```



Attention

Les attributs `headers` et `id` ne doivent pas être utilisés en combinaison de l'attribut `scope`.



Remarque

Une bonne pratique d'accessibilité consiste à respecter un ordre logique lorsque les valeurs des attributs `id` des cellules d'entêtes associées à une cellule de données sont intégrées dans l'attribut `headers` de cette dernière.

L'ordre de ces valeurs sera l'ordre de lecture par un lecteur d'écran.

10.1. Contenus masqués

10.1.1. Ne pas utiliser les propriétés CSS display ou visibility pour masquer un contenu qui ne peut pas être affiché par l'utilisateur

Lorsque des éléments sont présents dans le code HTML, mais ne sont pas affichés par défaut à l'écran, deux cas se présentent.

Cas 1 : ces éléments sont masqués mais peuvent être affichés sur action de l'utilisateur

Concernant le premier cas, il peut s'agir par exemple du contenu d'une pop-in, d'un menu déroulant, ou encore des différents panneaux d'un carrousel dynamique.

Pour savoir comment prendre en compte l'accessibilité de ces éléments, se référer à la notice d'accessibilité interfaces riches et JavaScript.

Cas 2 : ces éléments sont masqués mais ne seront jamais affichés

Les éléments masqués mais qui ne seront jamais affichés peuvent être également classés en deux sous-types.

Cas 2.1 : les éléments inutiles pour tous les utilisateurs

Les éléments inutiles pour tous les utilisateurs doivent, par ordre de priorité :

- Être supprimés du code HTML.
- Si cela n'est pas possible, être masqués à l'aide des propriétés CSS `display: none;` ou `visibility: hidden;`.

Cas 2.2 : les éléments qui sont utiles pour les utilisateurs de lecteurs d'écran, mais inutiles pour les autres utilisateurs

Les éléments utiles aux utilisateurs de lecteurs d'écran doivent quant à eux être rendus invisibles ou sortis de l'écran à l'aide des propriétés CSS `opacity: 0;` ou `position: absolute;` par exemple.

Il peut s'agir par exemple d'indications servant à faciliter la navigation dans les pages, de liens d'évitement, etc.

Attention

Lorsque des liens, ou plus généralement des éléments interactifs, sont intégrés dans un contenu destiné aux utilisateurs de lecteurs d'écran, utiliser `tabindex="-1"` sur ces éléments afin d'en empêcher l'accès par voie classique au clavier.

Par exemple :

```
<a href="alternative-carte-interactive.html" tabindex="-1"
class="ally">Utilisateurs de lecteurs d'écran, veuillez suivre ce lien pour
accéder à la version alternative de la carte interactive.</a>

.ally
{
position: absolute;
left: -999999px;
}
```

10.2. Modification de l'apparence et de la taille du texte

10.2.1. Utiliser CSS pour mettre en forme les textes

Hors logos et autres éléments distinctifs qui appartiennent à une charte graphique et qui ne sont pas à prendre en compte dans le cadre de cette recommandation, utiliser CSS pour mettre en forme les textes. N'utiliser des images-textes qu'en dernier recours.



Remarque

Une attention particulière a dû être portée sur la faisabilité de ce point dès la phase de conception graphique.

Si cela n'est pas le cas, se référer au critère « 2.1.3. Veiller à ce que les polices puissent être intégrées sous forme de texte » de la notice d'accessibilité pour la conception graphique et rentrer en contact avec la personne en charge de la réalisation des maquettes.



Astuce

Si nécessaire, ne pas hésiter à utiliser la propriété CSS3 `@font-face`.



Attention

Les technologies comme cufón ou sIFR ne permettent pas de répondre aux exigences en termes d'accessibilité et ne doivent pas être utilisées dans le cadre d'un site ou d'une application accessible.

10.2.2. Utiliser uniquement des tailles relatives (em, %, small, etc.) pour les polices de caractères

Pour définir les tailles des polices de caractères, utiliser uniquement des unités relatives pour la propriété CSS `font-size` comme par exemple `em`, `%`, `px`, `rem`, les mots-clés `x-small`, `small`, etc.

AcceDe Web – www.accede-web.com	Notice d'accessibilité HTML et CSS	Page 39/47
Réalisé par Atalan – accede@atalan.fr		Octobre 2012

Ne pas utiliser d'unités absolues comme pt, cm, etc.

Attention

De par son mauvais support dans Internet Explorer, l'utilisation du pixel (px) comme unité de taille de police est interdite dans les référentiels comme AccessiWeb ou RGAA. Pour garantir la conformité avec ceux-ci, il faudra utiliser une autre unité de taille pour la propriété font-size.

10.2.3. Garantir la lisibilité des contenus même lorsque la taille du texte est doublée

Garantir la lisibilité des contenus, même lorsque l'utilisateur double la taille du texte dans son navigateur. Éviter ainsi le chevauchement de texte, la disparition de texte de l'écran, etc.

Pour garantir au maximum le bon respect de cette recommandation, il est recommandé de :

- Ne pas utiliser d'unités (px, pt, %, em, etc.) avec la propriété CSS line-height.
- Ne pas définir de hauteur fixe sur les éléments susceptibles d'accueillir du contenu textuel, notamment les champs de formulaires.
- Ne pas définir de largeur en em sur les éléments.



Astuce

Pour tester ce critère :

- Avec **Firefox**, dans le menu *Affichage* puis *Zoom*, cocher *Zoom texte* seulement, puis doubler la taille du texte en faisant 6 fois le raccourci clavier **CTRL** et **+** (**CTRL** et **0** permet de revenir à la taille par défaut).
- Avec **Internet Explorer**, dans le menu *Affichage* puis *Taille du texte*, choisir *La plus grande*.

10.3. Accès à l'information

10.3.1. Garantir la lisibilité des contenus lorsque les images ne sont pas affichées

Lorsque les images ne sont pas affichées, les contenus textuels de la page doivent rester visibles et lisibles. Aucune information ne doit être perdue, et le contraste entre la couleur du texte et la couleur d'arrière-plan doit être suffisamment élevé (se référer à la notice d'accessibilité pour la conception graphique).

C'est-à-dire que tous les contenus doivent pouvoir être lus :

- Même lorsque les images intégrées en CSS ne sont pas chargées dans la page.
- Même lorsque les images intégrées en HTML sont remplacées par le contenu de leur attribut alt.

AcceDe Web – www.accede-web.com	Notice d'accessibilité HTML et CSS	Page 40/47
Réalisé par Atalan – accede@atalan.fr		Octobre 2012



Astuce

Chaque fois qu'un texte est superposé à une image d'arrière-plan, veiller à mettre en place une couleur de remplacement, qui garantira la lisibilité du texte en l'absence de cette dernière.

```
background: black url(..../images/fond-sombre.png) repeat-x;
```

Cette couleur de remplacement peut être héritée d'un élément parent.



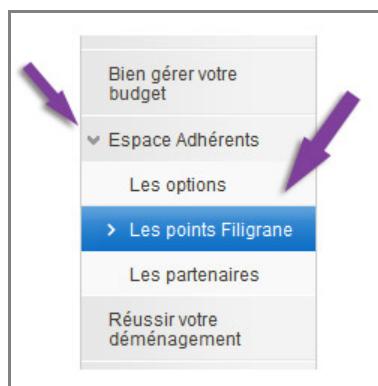
Attention

Une attention particulière doit être portée sur la lisibilité des alternatives textuelles des images intégrées dans le code HTML, lorsque les images ne sont pas affichées.

L'absence demandée dans une précédente recommandation de cette notice des attributs `height` et `width` dans le code HTML permet de garantir l'affichage des alternatives.

10.3.2. Garantir la compréhension de l'information même lorsque CSS est désactivé

Veiller à ce que l'information reste compréhensible même lorsque CSS est désactivé, notamment lorsque des couleurs, des tailles ou des positions sont vecteurs d'informations.



Dans cet exemple, une flèche tournée vers le bas marque l'élément déplié du menu (« Espace Adhérent ») tandis qu'un fond de couleur et une flèche tournée vers la droite marquent l'élément de la page courante (« Les points Filigrane »).

- [Bien gérer votre budget](#)
- [Espace Adhérents](#)
 - [Les options](#)
 - [Les points Filigrane](#)
 - [Les partenaires](#)
- [Réussir votre déménagement](#)

Lorsque CSS est désactivé, la flèche et les couleurs de fond disparaissent. Cependant, « Espace adhérent » étant la liste seule dépliée, et “Les points filigrane” le seul élément sans lien `<a>`, l'information reste compréhensible.

11.1. Ordre de tabulation

11.1.1. Veiller à ce que l'ordre de tabulation suive la logique de l'ordre de lecture

L'ordre de tabulation doit suivre la logique de l'ordre de lecture visuel de la page.

C'est-à-dire que lorsqu'un élément interactif en précède immédiatement un autre lors du parcours visuel de la page, le focus doit continuer sur le second élément immédiatement après avoir quitté le premier.

⚠️ Attention

Sauf cas très rares où l'interface est totalement stable et maîtrisée sur le long terme, veiller à ne jamais utiliser l'attribut `tabindex` avec une valeur supérieure à 0, au risque de perturber la logique de l'ordre de tabulation dans la page.

11.1.2. Mettre en place des liens d'évitement

Un lien d'évitement du type « Aller au contenu » doit systématiquement être présent sur chaque page afin de faciliter la navigation au clavier.

Ce lien doit être le premier élément interactif dans le code HTML. Il s'agit d'un lien vers une ancre qui doit permettre un accès direct au contenu principal de la page.

```
<a id="evitemen" href="#contenu">Aller au contenu</a>  
[ ... ]  
<div id="contenu" role="main">  
    [...]  
</div>
```



Attention

Bien qu'il soit fortement recommandé d'afficher ce lien, celui-ci peut être masqué par défaut, s'il n'a pas été prévu dans les maquettes graphiques. En revanche, il doit dans tous les cas être rendu visible à la prise de focus au clavier.

Le lien d'évitement ne doit jamais être masqué à l'aide des propriétés CSS `display: none;` et/ou `visibility: hidden;` sous peine de le rendre totalement inatteignable au clavier.

Privilégier une autre solution, par exemple l'utilisation des codes suivants :

```
a#evitement
{
position: absolute;
left: -999999px;
}

a#evitement:focus
{
position: static;
}
```



Remarque

Dans certaines situations, beaucoup de tabulations sont nécessaires pour accéder au menu principal et/ou au moteur de recherche depuis le sommet de la page.

Une bonne pratique d'accessibilité consiste alors à mettre en place une liste de liens d'évitement.

```
<ul id="evitement">
  <li><a href="#contenu">Aller au contenu</a></li>
  <li><a href="#menu">Aller au menu</a></li>
  <li><a href="#recherche">Aller à la recherche</a></li>
</ul>
```

11.2. Visibilité de la prise de focus

11.2.1. Garantir la visibilité de la prise de focus au clavier

Afin de permettre aux utilisateurs qui naviguent au clavier de se situer dans la page, chaque élément interactif doit être visuellement mis en avant lors de la prise de focus (liens, champs de formulaires, boutons, etc.).



Astuce

Une bonne pratique d'accessibilité consiste à doubler systématiquement chaque règle `:hover` par une règle `:focus` dans la CSS.



Attention

La règle `outline: none;` ne doit jamais être appliquée par défaut au focus dans la CSS, sauf dans les cas très précis où la visibilité du focus est ensuite surchargée plus tard dans la feuille de style.

Si les styles appliqués à la prise de focus sont jugés gênants lors de la navigation à la souris, il est possible de les annuler au clic par l'intermédiaire de la pseudo-classe `:active`.

Par exemple :

```
:focus  
{  
outline: 3px dotted green;  
}  
  
:active  
{  
outline: none;  
}
```

12.1. Contenus multimédia

12.1.1. Prévoir une alternative à chaque contenu multimédia (`<video>`, `<audio>`, `<object>`, etc.)

Prévoir une alternative à chaque contenu multimédia embarqué dans la page avec l'aide par exemple des balises `<video>`, `<audio>`, `<object>`, etc.

Cette alternative permettra de fournir le même niveau d'informations aux utilisateurs ne pouvant pas accéder au contenu multimédia (absence de plugin, navigateur vieillissant, etc.). Elle peut prendre la forme d'un contenu HTML, d'une image munie d'une alternative, etc.



Astuce

Lorsqu'une vidéo est affichée sur une page, sa transcription peut être considérée comme alternative. Il peut également s'agir de la vidéo en téléchargement (en supplément de la transcription).

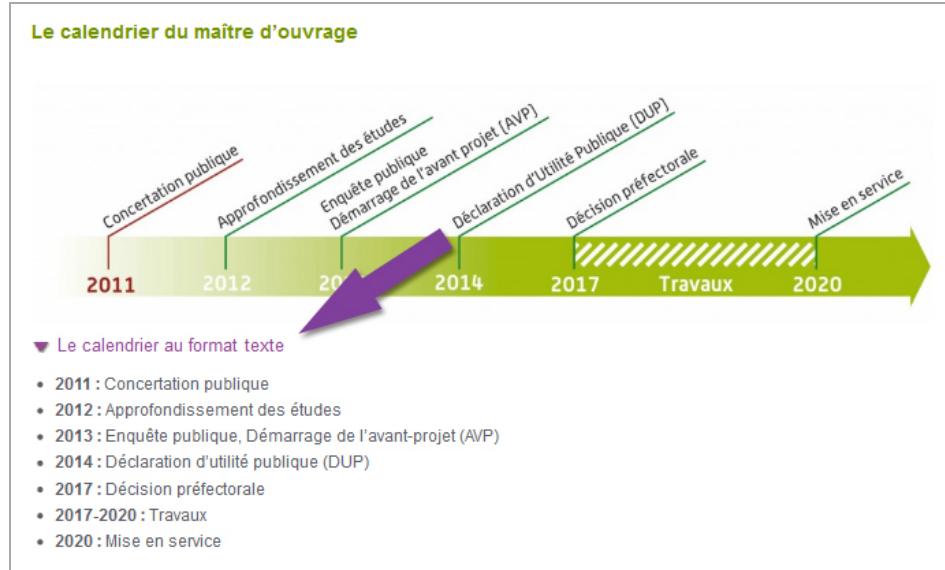


Figure 4 : le calendrier ci-dessus est une animation Flash porteuse d'informations, un lien « Le calendrier au format texte » permet à l'utilisateur d'afficher une alternative au format HTML.

12.1.2. Ne pas utiliser le paramètre wmode avec les valeurs transparent ou opaque

Lorsqu'une animation Flash est embarquée dans la page, le paramètre `wmode="transparent"` ou `wmode="opaque"` ne doit pas être utilisé, car susceptible de rendre les animations inutilisables au clavier.

Annexe 1 : recommandations additionnelles...

Dans un objectif d'être pragmatiques et directement opérationnels, plusieurs critères d'accessibilité présents dans les référentiels d'accessibilité n'ont pas été conservés dans les recommandations de cette notice. Il s'agit de critères rarement applicables dans un projet web ou de critères moins prioritaires. Ils sont cependant listés ci-dessous et seront détaillés ultérieurement sur le wiki du projet AcceDe Web².

...pour l'atteinte du niveau A (WCAG) ou Bronze (AccessiWeb)

- Baliser les citations en ligne avec `<q>`.
- Résumer le contenu et la structure de chaque tableau de données avec l'attribut `summary`.
- Renseigner l'attribut `title` sur chaque `<iframe>`.
- Renseigner l'attribut `title` sur chaque `<frame>`.
- Regrouper les options de même nature avec `<optgroup>` dans les `<select>`.
- Indiquer la langue de chaque document en téléchargement rédigé dans une langue étrangère.
- Doubler les images dotées d'un attribut `ismap` avec des liens alternatifs.
- Renseigner le sens de lecture principal de la page avec l'attribut `dir="ltr"` (gauche à droite) ou `dir="rtl"` (droite à gauche) sur la balise `<html>`.
- Utiliser l'attribut `dir` pour signaler les changements de sens de lecture dans le corps de la page.
- Identifier les principales zones de la page avec l'attribut `id` ou une ancre nommée.

...pour l'atteinte du niveau AA (WCAG) ou Argent (AccessiWeb)

Aucune recommandation additionnelle au niveau AA (WCAG) ou Argent (AccessiWeb).

...pour l'atteinte du niveau AAA (WCAG) ou Or (AccessiWeb)

- Prévoir des intitulés de liens explicites hors contexte et sans l'utilisation de l'attribut `title`.
- Baliser les définitions avec `<dfn>`.
- Baliser les abréviations avec `<abbr>`.
- Baliser les acronymes avec `<acronym>`.
- Limiter la largeur des textes à 80 caractères, quelle que soit la taille du texte.
- Prévoir un interligne d'au moins 1,5 fois la taille du texte, quelle que soit la taille du texte.
- Prévoir une marge entre les paragraphes d'au moins 1,5 fois la valeur de l'interligne, quelle que soit la taille du texte.
- Garantir l'absence de barre de défilement horizontale même lorsque la taille du texte est doublée par l'utilisateur.
- Garantir des longueurs de lignes inférieures à 80 caractères lorsque l'utilisateur réduit la largeur de la fenêtre du navigateur.

² <http://wiki.accede-web.com/notices/html-css/recommandations-additionnelles>

AcceDe Web – www.accede-web.com	Notice d'accessibilité HTML et CSS	Page 46/47
Réalisé par Atalan – accede@atalan.fr		Octobre 2012

Annexe 2 : grille de correspondances WCAG/AccessiWeb/RGAA

L'ensemble des recommandations présentées dans ce document est tiré des référentiels d'accessibilité WCAG, AccessiWeb et RGAA.

Afin d'obtenir un niveau de lecture selon les niveaux de ces référentiels (A, AA, AAA ou Bronze, Argent, Or) une grille de correspondance entre les recommandations HTML/CSS d'AcceDe Web et les critères de ces référentiels est maintenue sur le wiki du projet à l'adresse suivante : <http://wiki.accede-web.com/notices/html-css/grille-de-correspondances-wcag-accessiweb-rgaa>.

AcceDe Web – www.accede-web.com	Notice d'accessibilité HTML et CSS	Page 47/47
Réalisé par Atalan – accede @atalan.fr		Octobre 2012