Dernière mise à jour : 07/07/2016

GUIDE DE L'INTÉGRATEUR RGAA 3



Sommaire

Guide de l'intégrateur RGAA 3	
Licence	3
Introduction	5
À qui s'adresse ce guide ?	5
Mode d'emploi du guide	
Intégration web et RGAA 3	
Ressources connexes et références	
Fiche 1. Gabarit général	7
Introduction - cas utilisateurs	
L'essentiel	8
Éléments HTML5 et landmarks ARIA	9
Fiche 2. Navigation	13
Introduction - cas utilisateurs	
Liens d'accès rapide	14
2 systèmes de navigation au moins	
Plan du site	
Titre de pages	
Collection de pages	16
Page en cours de consultation dans le menu	17
Fil d'Ariane	17
Fiche 3. Contenus et structuration	18
Introduction - cas utilisateurs	18
Cadres	
Titres	19
Listes	20
Changement de langue	
Changement de sens de lecture	21
Nouvelles fenêtres	
Fichiers en téléchargement	22
Fiche 4. Tableaux	24
Introduction - cas utilisateurs	24
Tableaux de mise en forme	25
Tableaux de données simples	25
Tableaux de données complexes	27
Fiche 5. Liens	29
Introduction - cas utilisateurs	29
Liens explicites	30
Liens images	30
Liens vides	30
Titro do ligas titlo	21

Guide de l'intégrateur RGAA 3

Fiche 6. Formulaires	32
Introduction - cas utilisateurs	32
Étiquettes	32
Champs obligatoires	35
Aides à la saisie	35
Erreurs de saisie	36
Listes de choix	
Regroupement de champs	38
Fiche 7. La prise de focus et la navigation clavier	
Introduction - cas utilisateurs	39
Conserver l'outline	
Visibilité du survol et la prise de focus pour les liens	40
Ordre de tabulation cohérent	
Piège au clavier	41
Fiche 8. Respecter la distinction fond et forme	
Introduction - cas utilisateurs	
Utiliser les balises HTML pour leur sens propre	
Mise en forme exclusive via CSS	
Contenu visible sans CSS	
Contenu compréhensible sans CSS	47
Textes cachés correctement positionnés	48
Fiche 9. Images	49
Introduction - cas utilisateurs	49
Alternative obligatoire	50
Image de décoration : alternative vide	50
Image porteuse d'information : alternative renseignée	50
Images textes	51
Captcha	51
Image légendée	52
Fiche 10. Informations par la forme et la couleur	53
Introduction - cas utilisateurs	53
Informations par la couleur	53
Informations par la forme, taille ou position	54
Fiche 11. Agrandissement des caractères	55
Introduction - cas utilisateurs	55
Agrandissements des caractères en fonction du navigateur	55
Tailles de polices en unités relatives exclusivement	56
Lisibilité lorsque la taille de la police est agrandie	56
Convertir ses tailles de polices du pixel au em	
Responsive et agrandissement des caractères : une pierre, deux coups	58
Fiche 12. Multimédia	
Introduction - cas utilisateurs	59
Identifier la vidéo	59
Contenus multimédia non temporels	60

Licence

Ce document est la propriété du Secrétariat général pour la modernisation de l'action publique français (SGMAP). Il est placé sous <u>licence ouverte 1.0 ou ultérieure</u>, équivalente à une licence Creative Commons BY.

Vous êtes libres de :

- Reproduire, copier, publier et transmettre ces informations.
- Diffuser et redistribuer ces informations.
- Adapter, modifier, extraire et transformer ces informations, notamment pour créer des informations dérivées.
- Exploiter ces informations à titre commercial, par exemple en la combinant avec d'autres informations ou en l'incluant dans votre propre produit ou application.

Ces libertés s'appliquent sous réserve de mentionner la paternité de l'information d'origine : sa source et la date de sa dernière mise à jour.

Cette mention de paternité ne doit ni conférer un caractère officiel à la réutilisation de ces informations ni suggérer une quelconque reconnaissance ou caution par le producteur de l'information ou par toute autre entité publique, du réutilisateur ou de sa réutilisation.

Pour indiquer la paternité, ajouter un lien vers la version originale du document disponible sur le <u>compte</u> <u>GitHub de la DInSIC</u> (Direction interministérielle du numérique et du système d'information et de communication de l'État).

Introduction

Ce guide de l'intégrateur vous est proposé dans le cadre des ressources accompagnant la prise en main de la version 3 du référentiel général d'accessibilité pour les administrations (RGAA 3).

Toutes les règles et tous les exemples d'implémentation donnés ici se réfèrent à la version 3 du RGAA.

Le RGAA 3 est composé d'un <u>document d'introduction</u>, d'un <u>guide d'accompagnement</u> et d'un <u>référentiel technique</u>. Cet ensemble de documents a une portée réglementaire puisqu'ils ont été rendus officiels par l'<u>arrêté du 29 avril 2015</u>, lui-même venant préciser l'<u>article 47 de la loi 2005-102 du 11 février 2005</u> et l'arrêté 2009-546 du 14 mai 2009.

Les ressources complémentaires sont des supports sans valeur réglementaire et visent à vous aider à rendre vos contenus numériques accessibles et conformes au RGAA 3.

À QUI S'ADRESSE CE GUIDE ?

Ce guide s'adresse aux intégrateurs chargés de réaliser les gabarits HTML et CSS et se focalise sur les problématiques de respect de la sémantique et de la mise en forme. Il n'est pas question dans ce guide de JavaScript, qui fait l'objet avec ARIA, d'un tout autre guide.

À chacun son métier

Certaines problématiques ne sont volontairement pas abordées, car elles dépendent d'autres acteurs que l'intégrateur web. Par exemple, les contrastes de couleurs ou la pertinence des liens ne sont pas du ressort de l'intégrateur. Ce sont aux concepteurs, designers ou contributeurs de s'attacher à respecter ces contraintes.

MODE D'EMPLOI DU GUIDE

Le guide a été pensé comme une série de fiches pratiques sur le même modèle pour en faciliter l'utilisation. Vous pouvez les lire en fonction de vos besoins, sans avoir à suivre un ordre particulier, chaque fiche étant indépendante.

Certaines fiches proposent également des démonstrations d'implémentation conformes et non conforme. Les démonstrations, exemples et explications s'appuient uniquement sur les navigateurs utilisés dans <u>la</u> base de référence et sur les recommandations de la version 3 du RGAA.

Vous retrouverez dans chaque fiche :

- Synthèse : les principaux points à retenir ;
- Correspondances RGAA 3 : les correspondances avec les critères du référentiel technique du RGAA 3

INTÉGRATION WEB ET RGAA 3

Thématiques du RGAA 3 abordées à travers ces fiches :

- Images
- Cadres
- Tableaux
- Liens
- Éléments obligatoires
- Structuration de l'information
- Présentation de l'information
- Formulaires
- Navigation
- Consultation

RESSOURCES CONNEXES ET RÉFÉRENCES

- Guide « Contribuer sur le web de manière accessible »
- Méthodologie de test RGAA 3
- Comprendre les WCAG : description de cas utilisateurs et d'impacts par type de handicap
- Liste des techniques et échecs WCAG 2.0 (ressource en anglais)

Fiche 1. Gabarit général

INTRODUCTION - CAS UTILISATEURS

Dans cette fiche, nous expliquons la structure de base d'une page web et les éléments primordiaux que vous devez mettre en place avant toute autre intégration. C'est le squelette de votre site web. Vous verrez comment structurer votre gabarit avec les éléments HTML5 et les landmarks ARIA.

Les utilisateurs aveugles ne perçoivent pas les mises en forme. Pourtant, celles-ci aident à la compréhension de la fonction de certaines régions de la page.

Par exemple, on trouve généralement le menu de navigation sous forme horizontale avec une certaine mise en exergue, de façon à faire comprendre que cet élément permet de naviguer dans le site. Pour un utilisateur aveugle, cette navigation n'est rien d'autre qu'une liste de liens, comme il peut y en avoir d'autres dans la page.

Utiliser des marqueurs sémantiques (balises HTML5 et landmarks ARIA) va permettre d'enrichir la restitution pour ces utilisateurs : la navigation principale ne sera plus perçue simplement comme une liste de liens, elle sera restituée à l'utilisateur comme un élément de navigation, par l'intermédiaire du lecteur d'écran. L'utilisateur peut alors la distinguer des autres listes de liens.

De plus, ces marqueurs sémantiques vont également constituer des éléments de navigation rapide dans la page. Grâce à un raccourci clavier, certains utilisateurs vont pouvoir naviguer entre les régions que vous aurez identifiées.

SYNTHÈSE

- Déclarez un doctype valide.
- Déclarez toujours la langue de traitement sur la balise <html>.
- Veillez à toujours valider votre code source.
- Structurez votre page avec les landmarks ARIA.
- Si votre site est en HTML5, structurez votre page avec les balises HTML5 et les landmarks ARIA.

L'ESSENTIEL

Le DOCTYPE et la langue

Votre document doit débuter par une déclaration de doctype. Le W3C met à disposition <u>une liste des</u> doctypes valides utilisables sur le web.

Une fois cette déclaration faite, vous devez définir la langue de traitement principal de votre page. Si votre site est en français, alors vous devez déclarer du français.

```
<!DOCTYPE html>
<html lang="fr">
</html>
```

Un code source valide

Cette déclaration du type de document induit que vous respectez ensuite les normes d'écriture correspondantes, et notamment que votre code source est valide. Pour valider votre code source, utilisez toujours le code source généré (avec JavaScript) avec l'outil en ligne de validation du W3C.

Un code source valide est essentiel pour la robustesse du site et pour s'assurer d'un rendu homogène avec tous les navigateurs. En effet, un code mal structuré est généralement « réparé » par le navigateur. Mais tous les navigateurs ne réparent pas de la même manière, ce qui peut générer des différences importantes d'un navigateur à l'autre. Un code source valide est donc un prérequis essentiel, car il assure un code robuste et univoque.

L'outil en ligne sort deux types de résultats : les warnings et les errors. Seules les erreurs sont à prendre en compte. Le RGAA ne demande pas un code source valide à 100%.

Par exemple, les erreurs relatives à l'écriture des URL (notamment la présence d'esperluette dans les URL) ne sont pas considérées comme des erreurs d'accessibilité. Si ce sont les seules erreurs sur votre page web, alors votre code source est valide au regard du RGAA.

Pour le RGAA, un code source valide est un code source :

- dont les balises respectent les règles d'écriture (les balises sont conformes au type de document déclaré) ;
- odont l'imbrication des balises est conforme (par exemple, pas de lien imbriqué dans un autre lien);
- dont l'ouverture et la fermeture des balises sont conformes (par exemple, la règle « premier ouvert, dernier fermé » est respectée);
- dont les attributs respectent les règles d'écriture (par exemple, aucun élément obsolète n'est présent);
- dont les valeurs des attributs respectent les règles d'écriture (par exemple, les valeurs d'identifiant dupliquées ne sont pas conformes).

ÉLÉMENTS HTML5 ET LANDMARKS ARIA

À partir de la maquette graphique, avant de vous lancer dans l'intégration, observez et repérez :

- l'en-tête principal;
- le contenu principal;
- le pied de page (informations légales, etc.);
- le moteur de recherche;
- la navigation principale et les éventuelles navigations secondaires.

À chacune de ces zones correspond une propriété ARIA particulière qui va permettre de les identifier avec certitude. Ces propriétés s'implémentent avec l'attribut role :

```
l'en-tête principal : role="banner";
le contenu principal : role="main";
le pied de page (informations légales, etc.) : role="footer";
le moteur de recherche : role="search";
la navigation principale et les éventuelles navigations secondaires : role="nav".
```

Dans le cas où votre site est en HTML5, chacune de ces zones est identifiée par une balise HTML5 particulière :

```
l'en-tête principal : <header> ;
le contenu principal : <main> ;
le pied de page (informations légales, etc.) : <footer> ;
la navigation principale et les éventuelles navigations secondaires : <nav>.
```

De plus, afin de créer des points de navigation, la zone de contenu et la navigation principale (et les éventuelles navigations secondaires) doivent être identifiées, à l'aide de l'un des procédés suivants :

- un attribut id;
- une ancre précédant immédiatement la zone ;
- une ancre avant le premier élément de la zone.

Si votre site est en HTML5, votre structure de départ doit ressembler à ceci :

Si votre site utilise une autre déclaration de doctype (XHTML, HTML4.0...), vous devez simplement implémenter les landmarks ARIA sur les éléments correspondants. La structure doit ressembler à ceci :

Important :

- On peut trouver plusieurs balises header, mais le role="banner doit être unique dans la page.
- La balise main et le role="main" sont uniques dans la page.
- On peut trouver plusieurs balises footer, mais le role="contentinfo" doit être unique dans la page.
- On peut trouver plusieurs balises nav et plusieurs role="navigation".
- Le role="search" est unique dans la page.

Comment déterminer ce qu'est...

L'en-tête principal?

L'en-tête principal est généralement le titre de votre site, avec sa baseline, si elle existe. Très souvent, il s'agit du titre de niveau 1 de votre site (h1).

L'en-tête principal contient généralement peu d'éléments.

Le contenu principal?

Le contenu principal d'une page est le contenu d'intérêt. C'est le contenu qui donne un sens à votre page et sans lequel votre page est inutile.

Les navigations principales et secondaires ?

Pour rappel, le RGAA 3 demande à ce que le site possède deux systèmes de navigation parmi : une navigation principale, un plan du site et un moteur de recherche.

La navigation principale, généralement appelée le « menu », est un élément que l'on retrouve sur toutes les pages de votre site web. C'est celle que l'on retrouve en début de contenu pour atteindre les rubriques principales de votre site.

Très souvent, on rencontre également des navigations contextuelles dans des rubriques spécifiques. Ces éléments sont à considérer comme des navigations secondaires et doivent être implémentés de la même manière que la navigation principale.

Astuce:

Afin de permettre aux utilisateurs qui naviguent avec un lecteur d'écran de distinguer toutes les navigations que vous aurez définies, vous pouvez les titrer grâce à la propriété aria-label.

```
<nav role="navigation" aria-label="menu principal"></nav>
<nav role="navigation" aria-label="menu rubrique formations"></nav>
```

Le moteur de recherche?

Le moteur de recherche peut être une alternative intéressante pour les utilisateurs qui ne pourraient pas utiliser le menu ou le plan du site. Pour qu'il soit considéré comme une alternative, il faut néanmoins qu'il permette, à partir de ses résultats, d'atteindre n'importe quel contenu du site.

Ainsi, un moteur de recherche limité à un catalogue ne peut pas être considéré comme un système de navigation.

Le pied de page ?

Le pied de page est à considérer comme étant un élément qui apporte des informations sur le site. On va y placer généralement, si elles sont présentes, les informations légales (copyright, lien vers les mentions légales...), les informations de contact (adresse, téléphone...).

La région de pied de page n'est pas nécessairement équivalente au pied de page de la maquette graphique.

Souvent, dans le pied de page des maquettes de site web, on trouve des éléments comme la répétition de la navigation (sorte de plan du site en pied de page). Cet élément n'est pas à inclure dans le pied de page (footer role="contentinfo").

Pourquoi doubler les balises HTML5 avec les landmarks ARIA?

Ajouter les landmarks ARIA aux balises HTML5 correspondantes permet de distinguer les autres éléments du même type qui pourraient être utilisés.

Vous pourriez mettre plusieurs balises header ou footer dans votre structure de site. Ces balises pouvant être contenus dans des balises article ou section pour créer des en-têtes d'articles et des

pieds d'article. Ajouter les landmarks permet de différencier toutes ces balises des balises structurant le corps principal de la page.

Le validateur ressort une alerte (warning), mais ce n'est pas une erreur. L'alerte correspond simplement au fait que les balises HTML5 ont déjà des rôles implicites correspondant à ces propriétés (banner est le rôle implicite de la balise header). L'alerte vous signifie simplement que l'utilisation de ces rôles est peut-être inutile. Néanmoins, pour les raisons évoquées au-dessus, la position du RGAA 3 reste stricte à ce sujet.

Vous pouvez consulter à ce sujet les rôles ARIA implicites sur les éléments HTML.

Utilisation optimale

Afin de rendre l'utilisation des régions (balise HTML5 et landmarks) optimale, il est toujours préférable de ne pas inclure dans l'en-tête la navigation principale par exemple. De manière générale, il est préférable de ne pas imbriquer les régions principales définies ici.

POUR ALLER PLUS LOIN

- Explications détaillées des éléments de structure dans la spécification HTML5 du W3C
- Les rôles ARIA implicites sur les éléments HTML

VOIR AILLEURS / RESSOURCES

- Liste des doctypes par le W3C
- Validateur de code du W3C
- <u>Voir la fiche sur les contenus, principalement la section sur les changements de langue pour une explication des codes de langue</u>

CRITÈRES RGAA

- 8.1 [A]
- 8.2 [A]
- 8.3 [A]
- 8.4 [A]
- 9.2 [A]
- 12.10 [A]

Fiche 2. Navigation

INTRODUCTION - CAS UTILISATEURS

Vous devez fournir aux utilisateurs au moins deux systèmes de navigation afin de couvrir un plus large spectre de stratégie de navigation dans les contenus du site. En effet, tous les utilisateurs ne naviguent pas de la même manière et certains systèmes de navigation peuvent s'avérer trop complexes à utiliser. Par exemple, un menu de navigation trop riche, comme un méga-menu, peut nécessiter beaucoup d'actions pour un utilisateur qui ne navigue qu'au clavier et donc rendre sa recherche d'information laborieuse. Proposer un second système permet d'aider ces utilisateurs qui ont des stratégies de navigation différentes.

Certains utilisateurs peuvent avoir des difficultés à atteindre la zone de contenu principal. Par exemple : les utilisateurs malvoyants qui utilisent une loupe d'écran ne voient pas la page dans sa globalité. Il peut être laborieux pour ces utilisateurs de devoir scroller tous les éléments communs des pages d'un site (l'en-tête, la navigation) pour atteindre le contenu principal à chaque fois. De même, les utilisateurs aveugles qui ont une lecture linéaire de la page doivent parcourir tous les éléments redondants (titre principal, navigation, etc.) avant d'atteindre le contenu principal.

Les utilisateurs qui ne peuvent pas utiliser de souris ou un système de navigation adapté restent néanmoins ceux qui rencontrent le plus de difficultés. Pour ces utilisateurs, la zone de contenu principal, comme tous les autres éléments du site ne peut être atteinte qu'avec la tabulation ce qui peut être particulièrement laborieux.

Les liens d'accès rapide en début de contenu leur permettent de réduire considérablement leur temps de consultation, ils peuvent directement atteindre la zone de contenu principal.

SYNTHÈSE

- Définir des titres de pages pertinents.
- Implémenter un lien d'accès rapide au contenu.
- Implémenter un lien d'accès rapide à la navigation principale.
- Implémenter des liens d'évitement pour tous les groupes de liens importants.
- Implémenter deux systèmes de navigation au moins.

LIENS D'ACCÈS RAPIDE

Vous devez fournir aux utilisateurs des liens d'accès rapide. Les liens d'accès rapide sont des ancres vers certains éléments de la page. Ainsi sur votre site, vous devez implémenter :

- un lien qui permet d'éviter ou d'accéder à la zone de contenu principale;
- un lien qui permet d'éviter ou d'accéder à chaque groupe de liens importants (notamment la navigation).

Ces liens d'accès rapide (ou d'évitement) doivent de préférence toujours être visibles. En effet, ils sont utiles aux utilisateurs aveugles, aux utilisateurs qui naviguent au clavier, mais également aux utilisateurs malvoyants qui utilisent une loupe d'écran. Ces derniers naviguent à la souris et si ces liens sont positionnés hors écran, il leur est impossible de les utiliser.

À défaut, si vous devez les positionner hors écran, vous devez au moins vous assurer que ces liens apparaissent à la prise de focus. Vous devez donc les positionner hors écran (position: absolute). N'utilisez surtout pas de propriétés CSS qui les rendent inactifs, telles que :

```
display: none; ;
visibility: hidden; ;
width et height avec les valeurs 0 (width: 0; height: 0;);
font-size: 0;;
attribut HTML5 hidden;
propriété aria-hidden="true".
```

Pour qu'ils soient efficaces, les liens d'accès rapide au contenu et à la navigation doivent être positionnés en début de code et en haut de page. En faire les premiers liens du code de votre page est la meilleure solution.



Important :

Ces liens doivent toujours être situés à la même place dans la présentation. De plus, afin que les utilisateurs aveugles, qui ont une lecture linéaire de votre site, puissent s'y retrouver, ces liens doivent toujours être dans le même ordre dans le code source.

Page de démonstration des cas d'implémentation des liens d'accès rapide

2 SYSTÈMES DE NAVIGATION AU MOINS

Un site web doit proposer deux systèmes de navigation au moins, parmi :

- une navigation principale;
- un moteur de recherche ;
- un plan du site.

Pour que le moteur de recherche puisse être considéré comme un élément de navigation, il doit indexer toutes les pages du site. Ainsi, un moteur de recherche dans un catalogue qui ne permet pas de trouver la page des mentions légales du site n'est pas considéré comme un moteur de recherche.

Le plan du site quant à lui doit être représentatif de la structure générale du site. Il n'est pas demandé que soient référencées toutes les pages du site, mais seulement les pages menant aux rubriques principales. Par exemple, si le site dispose d'une rubrique « Actualités », il n'est pas nécessaire d'y faire figurer toutes les actualités publiées.

Il existe des cas particuliers pour lesquels cette recommandation est non applicable :

- Un site composé d'un faible nombre de pages (2 à 3 pages) peut ne fournir qu'un menu de navigation menant vers ces pages, le menu étant alors identique au plan du site.
- Un site monopage (« Single Page Application ») peut ne fournir que des liens d'accès rapide aux différentes sections de la page, la fonction de recherche du navigateur (Ctrl F) permettant de chercher dans tout le contenu du site.

Ordre cohérent et présentation cohérente

Chaque système de navigation doit avoir sur chaque ensemble de pages un ordre et une présentation cohérente :

- le système de navigation est accessible depuis une fonctionnalité identique (par exemple, pour le plan du site, la fonctionnalité est un lien ayant pour intitulé « plan du site ») ;
- la fonctionnalité est située à la même place dans la présentation (par exemple, le lien vers le plan du site est toujours situé en haut à gauche sur la page) ;
- la fonctionnalité est toujours dans le même ordre relatif dans le code source (par exemple, le lien vers le plan du site est toujours le premier élément texte et le premier lien de la page).

PLAN DU SITE

Le plan du site n'est pas obligatoire, il fait partie des 3 systèmes de navigation reconnus. Si vous intégrez un plan de site, vous devez vous assurer en plus que tous les liens sont pertinents et fonctionnels. <u>Vous pouvez utiliser l'outil de validation des liens du W3C</u> pour vous en assurer.

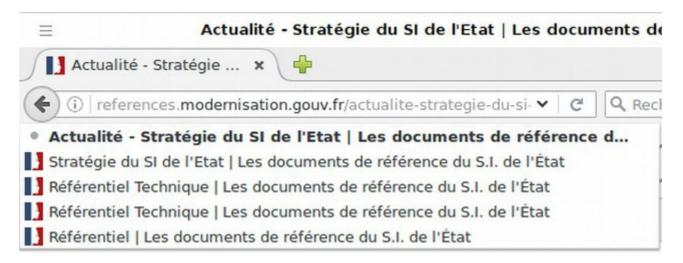
TITRE DE PAGES

Le titre des pages (balise <title>) est un élément de repère pour nombre d'utilisateurs.

Pour les personnes aveugles, utilisateurs de lecteurs d'écran, le titre de la page est le premier élément vocalisé. Pour des utilisateurs qui naviguent avec l'historique de navigation ou la liste des onglets, il est important qu'ils puissent retrouver les pages facilement en se basant sur leur titre.

Dans un certain nombre de cas, le titre de page est simple à construire, il reprend souvent le titre principal du contenu et le nom du site. Très souvent, ce n'est pas vous qui le créez, on vous impose sa construction (conception, référencement...). Dans d'autres cas, notamment pour les interfaces interactives, il faut veiller à rendre ces titres explicites :

- de collections de pages (résultats de recherche par exemple). Dans ces cas, il est important d'indiquer la pagination dans le titre ;
- lorsqu'un formulaire renvoie des erreurs, le titre de la page doit le mentionner.



COLLECTION DE PAGES

Une collection de pages est un ensemble de pages reliées les unes aux autres par des liens et ayant un thème ou une nature commune. Par exemple, les pages de résultats d'un moteur de recherche ou les pages d'un catalogue sont des collections de pages.

Pour les pages d'une collection de pages, vous devez mettre à disposition de l'utilisateur :

- un lien permettant d'accéder à la page suivante;
- un lien permettant d'accéder à la page précédente;
- des liens permettant d'accéder à toutes les pages de la collection.

Lorsque le nombre de pages est très important, il est illusoire de vouloir mettre un lien vers chaque page. Dans ce cas, il est acceptable de présenter les liens par groupe de 10 liens par exemple (« 20-30 »).

D'autres systèmes permettant d'atteindre une page quelconque d'une collection de pages peuvent être utilisés comme un champ de formulaire ou une liste select de navigation.



Important:

Une navigation dans une collection de pages est un élément de navigation, vous devez donc l'implémenter dans un élément <nav> avec le role="navigation". Et afin de distinguer les éléments de navigation, vous pouvez les labelliser au moyen de la propriété aria-label

```
<nav role="navigation" aria-label="navigation dans la collection de pages">
  [...]
</nav>
```

Page de démonstration des cas d'implémentation de la navigation dans les collections de pages

PAGE EN COURS DE CONSULTATION DANS LE MENU

Cette recommandation est de niveau AAA.

Indiquer la page en cours de consultation aux utilisateurs permet de donner des points de repère quant à leur localisation dans le site.

Lorsque vous indiquez la page en cours dans le menu de navigation, attention aux informations par la couleur et par la forme. Si vous décidez de mettre une couleur de fond différente au lien actif, n'oubliez pas d'ajouter un title sur le lien sur le modèle « [Titre de la page] en cours de consultation ».

La meilleure des implémentations consiste à couvrir la plus grande part d'utilisateurs, et donc :

- Utiliser une couleur de fond pour signifier la page active.
- Utiliser une forme adjacente au lien, par exemple une icône (pour les utilisateurs qui perçoivent mal ou pas les contrastes de couleurs).
- Ajouter un title sur le lien de la page active (pour les utilisateurs aveugles ou les utilisateurs qui utilisent des feuilles de styles utilisateurs).



FII D'ARIANE

Cette recommandation est de niveau AAA.

Un fil d'Ariane est un moyen de navigation rapide et un repère dans le site, il doit permettre à l'utilisateur de s'orienter dans le site web.

Le fil d'Ariane doit donc être représentatif de la position de la page dans l'arborescence du site. Par exemple, les pages parentes doivent y être représentées.

VOIR AILLEURS / RESSOURCES

• Outil en ligne mis à disposition par le W3C permettant de vérifier que les liens d'une page sont fonctionnels

CRITÈRES RGAA

- 8.5 [A]
- 8.6 [A]
- 12.1 [AA]
- 12.2 [AA]
- 12.3 [AA]
- 12.4 [AA]
- 12.5 [AA]

- 12.6 [AA]
- 12.7 [AA]
- 12.8 [AAA]
- 12.9 [AAA]
- 12.11 [A]
- 12.12 [AAA]

Fiche 3. Contenus et structuration

INTRODUCTION - CAS UTILISATEURS

Dans cette fiche, nous allons traiter des éléments de structure utilisés notamment dans les contenus et les corps de page.

Structurer les contenus avec les éléments appropriés offre une expérience de navigation enrichie pour les utilisateurs de lecteurs d'écran. Un utilisateur aveugle, malvoyant ou ayant des difficultés de lecture et qui utilise un lecteur d'écran pour parcourir le contenu d'une page, va utiliser des raccourcis clavier pour naviguer entre les types d'éléments. Il peut naviguer de titre en titre, aller au prochain champ de formulaire, afficher la liste des iframe ou encore aller à la liste précédente.

En plus de permettre de naviguer entre éléments, cela lui permet également de les éviter.

SYNTHÈSE

- Donnez un titre pertinent aux cadres en ligne.
- Utilisez les balises appropriées pour structurer les listes.
- Indiquez les changements de langue et de sens de lecture.
- Indiquez les ouvertures de nouvelles fenêtres.
- Indiquez le format et le poids des documents en téléchargements.

CADRES

Pour un utilisateur déficient visuel utilisant un lecteur d'écran, être prévenu du type de contenu d'un cadre lui permet de choisir s'il va l'explorer. De plus, avoir à disposition un titre fournit à l'utilisateur un moyen supplémentaire de navigation entre les éléments : il va pouvoir rechercher entre tous les cadres, ou aller de cadre en cadre.

Lorsque vous embarquez des contenus issus d'autres sites web via des cadres en ligne, vous devez implémenter un attribut title sur ces cadres qui permettent d'identifier le contenu. Les utilisateurs aveugles sont les plus impactés par cette problématique. Vous devez leur donner les indications suffisantes pour qu'ils sachent quels types de contenus ils vont y trouver. En effet, certains lecteurs d'écran offrent une fonctionnalité permettant d'afficher la liste des cadres de la page, le title servira à identifier les cadres dans cette liste.

<iframe title="Fil Twitter de la DInSIC"></iframe>

Dans le cas des cadres techniques, non visibles et utiles uniquement à la gestion des données par les scripts, un titre générique sur le modèle « Cadre technique » est suffisant.

TITRES

Utiliser des titres et des sous-titres permet de structurer et découper un texte. Cela fournit à l'utilisateur un plan du document et lui permet de naviguer de titre en titre pour se déplacer plus rapidement dans le contenu de la page.

Le titrage des contenus est une étape importante dans la structuration des contenus qui répond à deux besoins : identifier rapidement un contenu recherché et, surtout, naviguer rapidement dans le contenu en se déplaçant de titre en titre.

Il y a 3 obligations pour le titrage des contenus :

- la page doit comporter au moins un titre <h1>;
- la hiérarchie de titres doit être cohérente;
- tous les titres nécessaires doivent être présents.

Une hiérarchie cohérente est une hiérarchie qui ne contient pas de saut dans les niveaux de titres. Par exemple, après un h2 on doit trouver un h3 ou un autre h2, mais surtout pas un h4.

Enfin, tous les titres nécessaires à l'architecture de l'information doivent être présents. Il s'agit de proposer un plan de la page permettant de comprendre et de naviguer rapidement dans le contenu.

Astuce :

Pour valider la structure de votre page, vous pouvez utiliser l'<u>extension Firefox HeadingsMap</u>. Sélectionnez l'onglet « Headings » et vérifiez la cohérence et l'imbrication des titres.

ARIA

ARIA propose des propriétés qui permettent de transformer n'importe quel élément HTML en titre : role="heading" et aria-level="n".

Toutefois, certaines fonctionnalités de navigation rapide dans les titres peuvent ne plus fonctionner avec une implémentation de titre ARIA, qui ne devrait être utilisé que lorsque le recours à un vrai titre HTML n'est pas possible.



A Note:

Les outils de tests automatiques comme HeadingsMap ne font pas entrer les titres créés avec des propriétés ARIA dans les plans de documents qu'ils génèrent. Attention donc à vérifier manuellement vos hiérarchies si vous utilisez ce type d'implémentation.

LISTES

Pour un utilisateur aveugle naviguant avec un lecteur d'écran, le nombre d'éléments que contient la liste est annoncé, ainsi que le niveau de l'élément dans la liste. Ainsi, si l'utilisateur juge la liste trop longue, il pourra tout simplement l'éviter plutôt que de la parcourir en entier, il peut aussi aller au début de la liste, ou à la fin directement.

Toutes les listes doivent être structurées avec les balises appropriées.

On distingue 3 types de listes :

- les listes non ordonnées ul li ;
- les listes ordonnées ol li;
- les listes de définitions dl dd dt.

Toutes les suites d'éléments doivent être considérées comme une liste. Par exemple, une succession de liens, une succession d'indications courtes (recette de cuisine par exemple). Ces structures de listes sont surtout utiles aux utilisateurs aveugles pour qui la liste, la nature de la liste, le nombre d'éléments et le niveau de l'élément en cours de lecture sont vocalisés. De plus, grâce aux flèches de direction, ces utilisateurs peuvent circuler d'item de liste en item de liste ou encore, à l'aide d'un raccourci clavier, se déplacer rapidement de liste en liste.

Dans certains cas néanmoins, le recours à une liste peut-être considéré comme inutile. Par exemple, pour un bloc d'adresse isolé dans un contenu le recours à une liste n'a pas d'utilité pratique.

ARIA

ARIA propose des propriétés qui permettent de transformer une succession d'éléments en liste non ordonnée: role=list et role=list-item.

Toutefois, certaines fonctionnalités de navigation rapide dans les éléments de listes peuvent ne plus

fonctionner avec une implémentation de liste ARIA qui devrait être réservée lorsqu'il n'est pas possible d'utiliser une vraie liste HTML.

```
<div role="list">
  Accueil
  Actualités
  Contact
  <div>
```

CHANGEMENT DE LANGUE

Pour tous les termes de langue étrangère qui n'appartiennent pas au dictionnaire de la langue française (par exemple : brainstorming) et qui ne sont pas des noms propres (par exemple Watson), vous devez indiquer un changement de langue avec l'attribut lang sur l'élément lui-même ou un de ses éléments parents.

```
Retrouvez vos interprètes préférés au cours de l'<span
lang="en">after</span> organisé par nos partenaires.
```

L'indication des changements de langue est nécessaire pour indiquer aux technologies d'assistance de modifier la restitution vocale d'un élément.

Les changements de langue concernent tous les contenus, y compris les valeurs de certains attributs comme title. Il n'est pas possible d'indiquer des changements de langue dans une valeur d'attribut ellemême, dans ce cas le changement de langue est indiqué sur l'élément qui contient l'attribut. Un lien affecté d'un title en anglais devra comporter un attribut lang="en". Lorsque l'attribut contient plusieurs passages de texte dans des langues différentes, il est admis qu'aucune indication pertinente ne peut être faite et donc qu'aucun attribut ne soit présent.

La valeur de l'attribut lang correspond au code ISO de 2 caractères (ISO 639-1) ou 3 caractères (ISO 639-2 et suivants).

CHANGEMENT DE SENS DE LECTURE

Pour les changements de sens de lecture, vous devez implémenter l'attribut dir sur l'élément lui-même ou un parent pour traiter des contenus successifs. L'attribut dir prend une des valeurs suivantes :

- 1tr: left to right, de gauche à droite
- rt1 : right to left, de droite à gauche

```
<span lang="ar" dir="rtl">
شكرا جزيلا
«/span>
```

NOUVELLES FENÊTRES

Les ouvertures de nouvelles fenêtres peuvent demander beaucoup d'efforts à certains utilisateurs pour revenir à la page qu'ils consultaient. Ainsi, un utilisateur naviguant uniquement à l'aide de la touche tabulation va devoir effectuer des manipulations supplémentaires pour fermer et revenir à la fenêtre

Guide de l'intégrateur RGAA 3

d'origine. Les utilisateurs ayant des troubles de l'attention peuvent également être désorientés par les ouvertures de nouvelles fenêtres, surtout si elles ne sont pas de leur fait (ouverture sans action de l'utilisateur).

Pour chaque élément interactif qui ouvre une nouvelle fenêtre (ou un nouvel onglet), vous devez avertir l'utilisateur.

Sont concernés:

- les liens qui possèdent l'attribut target (avec n'importe quelle valeur sauf _self, _top ou _parent);
- les scripts qui déclenchent l'ouverture de nouvelles fenêtres;
- les éléments object ou embed qui ouvrent une nouvelle fenêtre ;
- les contrôles de formulaires (button, input type="submit", input type="reset" par exemple) qui déclenchent l'ouverture de nouvelles fenêtres.

Pour avertir l'utilisateur, vous pouvez :

- ajouter la mention « nouvelle fenêtre » directement dans l'intitulé de l'élément concerné;
- ajouter, si l'élément le permet, un title sur le modèle : Intitulé du lien nouvelle fenêtre (attention à respecter les règles de création des title).

Pas de nouvelles fenêtres sans action de l'utilisateur

L'ouverture d'une nouvelle fenêtre lorsque l'utilisateur ne l'a pas sollicitée peut être très perturbante.

Si cet usage est fortement déconseillé, il y a un cas où la gêne occasionnée est trop importante pour être accepté. Cela concerne l'ouverture de nouvelles fenêtres au chargement de la page. En effet, dans ce cas les personnes aveugles, utilisatrices de lecteur d'écran, et les personnes handicapées mentales peuvent ne plus savoir quelle est la page de contenu principal.



Important :

L'ouverture de nouvelles fenêtres au chargement de la page est strictement interdite.

FICHIERS EN TÉLÉCHARGEMENT

Sur le même principe que l'ouverture de nouvelles fenêtres, pour tous les documents en téléchargements, vous devez indiquer :

- le poids du document ;
- le format (PDF, doc, xls...);
- la langue si elle est différente de celle du site.

Cette information peut être mise à disposition dans le contexte adjacent du lien ou de l'élément de formulaire (par exemple la phrase). Il est néanmoins préférable d'ajouter ces informations sur l'élément lui-même, par exemple l'intitulé ou le title.

```
Téléchargez le <a href="#" title="catalogue 2016 (PDF, 450ko)">catalogue 2016</a>
```

Note:

Lorsque les documents en téléchargement sont générés à la volée (par exemple, le résultat d'un questionnaire ou d'une recherche) et qu'il n'est a priori pas possible de définir le poids du document, il est admis que seul le format soit indiqué.

VOIR AILLEURS / RESSOURCES

- Code de langue ISO 639-1
- La spécification ARIA au sujet du role="list"
- HeadingsMap, extension Firefox pour contrôler le plan du document

CRITÈRES RGAA

- 2.1 [A]
- 2.2 [A]
- 8.7 [AA]
- 8.8 [AA]
- 8.10 [A]
- 9.1 [A]
- 9.3 [A]
- 13.2 [A]
- 13.3 [A]
- 13.6 [A]

Fiche 4. Tableaux

INTRODUCTION - CAS UTILISATEURS

Un tableau est toujours complexe à utiliser avec un lecteur d'écran. Une personne aveugle n'a pas une vision globale du tableau. Elle le parcourt de manière séquentielle : une case après l'autre et ligne à ligne. Afin d'améliorer son expérience, les lecteurs d'écran offrent des fonctionnalités de parcours étendues en utilisant les flèches de direction.

Pour adapter le tableau aux capacités de restitution et aux fonctionnalités de navigation étendues, il est important de définir des en-têtes de colonnes ou de lignes et de les relier explicitement aux cellules de données.

SYNTHÈSE

- Implémenter le role="presentation" sur les tableaux de mise en forme.
- Déclarer les cellules d'en-têtes dans des éléments .
- Implémenter un attribut scope="col" pour les en-têtes de colonnes.
- Implémenter un attribut scope="row" pour les en-têtes de lignes.
- Utiliser les relations id, headers pour relier les cellules de données à leurs en-têtes dans le cas de cellules fusionnées.

TABLEAUX DE MISE EN FORME

Le premier conseil serait simplement de ne pas utiliser de tableau () pour réaliser des mises en forme. Les tableaux sont utiles pour présenter des données tabulaires. C'est leur fonction principale.

Si malgré tout, vous êtes amenés à utiliser des éléments <table pour réaliser certaines mises en forme, alors vous devez :

- Implémenter role="presentation" sur l'élément .
- Ne pas utiliser les éléments propres aux tableaux de données : <caption>, , , <thead>, <tfoot> ;
- Ne pas utiliser les attributs propres aux tableaux de données : scope, headers.

En résumé, si vous utilisez un tableau de données, la structure doit ressembler à celle-ci :

De plus, vous devez vous assurer que la linéarisation du tableau permet la compréhension du contenu : un tableau est lu de gauche à droite. Assurez-vous que le contenu reste compréhensible de cette manière.

Note:

L'API ARIA propose un mécanisme permettant de surcharger le rôle natif d'un élément pour proposer des composants. Ainsi, il est possible d'utiliser des tableaux de mise en forme pour construire des listes, par exemple en implémentant les rôles list et listitem sur les éléments du tableau. Si cet usage est fortement déconseillé, il est néanmoins conforme. Le tableau étant restitué comme une liste et non plus comme un tableau, il n'est pas utile de signaler qu'il s'agit d'un tableau de présentation via le rôle presentation.

À propos du role="presentation"

Le role="presentation" permet d'annuler la sémantique native de l'élément qui la possède.

Dans notre cas, si le tableau possède le role="presentation", un lecteur d'écran ne restituera pas le tableau, mais uniquement les contenus dans l'ordre dans lequel ils sont insérés dans les cellules. L'utilisateur aura l'impression d'être sur du simple texte.

De plus, les comportements clavier disponibles sur ces éléments (par exemple **flèche droite** pour aller à la cellule suivante) ne sont plus actifs et retrouvent leur fonctionnalité d'origine (par exemple, **flèche droite** permet de se déplacer de caractère en caractère).

Voir l'explication du role="presentation" dans la spécification WAI-ARIA

TABLEAUX DE DONNÉES SIMPLES

Un tableau est considéré comme simple s'il ne possède pas de cellules fusionnées qui rendent la compréhension délicate et s'il ne présente pas une structure (notamment des sous-contextes) qui rend sa compréhension dépendante d'une mise en forme.

Une manière d'identifier un tableau de donnée simple est de considérer les en-têtes : si les en-têtes concernent toute une ligne ou toute une colonne alors il s'agit d'un tableau de données simple.

Pour un tableau de données simple vous devez :

- Implémenter un élément <aption> qui est le titre du tableau, il permet aux utilisateurs de comprendre quelles données ils vont rencontrer.
- Implémenter les en-têtes dans des éléments ;
- Pour les en-têtes de colonnes, l'élément doit avoir l'attribut scope="col" ;
- Pour les en-têtes de lignes, l'élément doit avoir l'attribut scope="row".

Un tableau simple ressemble à ceci :

Accessibilité des lignes du réseau de surface RATP

Équipements	Nombre de lignes	Pourcentage du total de lignes
Rampe d'accès	260	70,46 %
Annonce sonore	318	86,17 %

La structure de notre tableau sera donc la suivante :

Ces données sont issues <u>du site data.gouv.fr</u>. Leur exploitation dans ce contexte n'est faite qu'à titre d'illustration, elle ne relève en rien d'une statistique officielle.

TABLEAUX DE DONNÉES COMPLEXES

Les tableaux de données complexes sont généralement des tableaux qui possèdent des cellules fusionnées ou des sous-contextes.

Voici un exemple de tableau de données complexe :

Nombre de gares et points d'arrêts TER, en fonction du type de handicap et du type de dispositif par handicap

Malvoyants		Aveugles	
Obstacles contrastés	Portes contrastées	Obstacles détectables à la canne	Guidage en braille
73	65	103	0
138		103	

Dans ce tableau, on trouve des en-têtes et des cellules de données fusionnées. La réparation consiste à relier chaque cellule de données explicitement grâce à des identifiants sur les en-têtes et des attributs headers sur les cellules de données.

Ces données sont issues du site <u>data.gouv.fr</u>. Leur exploitation dans ce contexte n'est faite qu'à titre d'illustration, elle ne relève en rien d'une statistique officielle.

POUR ALLER PLUS LOIN

• Le role="presentation" dans la spécification WAI-ARIA

VOIR AILLEURS / RESSOURCES

• <u>Fiche « Des tableaux pour présenter des données » dans le guide « Contribuer sur le Web de manière accessible ».</u>

CRITÈRES RGAA

- 5.1 [A]
- 5.2 [A]
- 5.3 [A]
- 5.4 [A]
- 5.5 [A]
- 5.6 [A]
- 5.7 [A]
- 5.8 [A]

Fiche 5. Liens

INTRODUCTION - CAS UTILISATEURS

Pour un utilisateur présentant une déficience visuelle et qui n'a pas une vision globale de la page, il est essentiel de comprendre la fonction d'un lien. En effet, le contexte visuel et la proximité d'information ne sont pas toujours disponibles pour ces utilisateurs aveugles ou malvoyants.

Pour des utilisateurs présentant des déficiences cognitives, des liens peu explicites n'incitent pas à l'action. L'utilisateur ne sait pas et ne comprend pas vers quelle page il se dirige si l'intitulé du lien ne l'explique pas suffisamment.

Lorsque les liens sont constitués par des images seules, c'est le texte alternatif de l'image qui constitue l'intitulé du lien. À ce moment, le texte alternatif à renseigner n'est plus la description de l'image, mais la destination du lien. Si l'image n'a pas d'alternative, le lien est alors un lien vide. Un utilisateur qui navigue à la voix ne pourra par exemple pas cliquer sur ce lien, ne sachant pas comment le nommer.

SYNTHÈSE

- Attention aux liens vides.
- Définissez des intitulés de liens pertinents.
- Respectez la construction des titres de liens.

LIENS EXPLICITES

Nous ne détaillerons pas ici les liens explicites, puisqu'ils ne sont généralement pas du ressort de l'intégrateur, mais plutôt du concepteur, du designer et du contributeur. Nous vous renvoyons à la <u>fiche</u> « <u>Créer du lien sans perdre son chemin » du quide « Contribuer sur le web de manière accessible ».</u>

À titre d'information, un lien doit être explicite par lui-même (son intitulé) ou grâce à son contexte dont les principaux sont :

- le titre (hx) précédant le lien ;
- le paragraphe (p) dans lequel est inséré le lien;
- la liste (ul ol dl) dans laquelle est inséré le lien;
- la ligne dans laquelle est inséré le lien.

Lorsqu'aucun de ces contextes ne permet de rendre le lien explicite, le recours au title, dont l'utilisation est détaillée plus bas, est la dernière solution possible.

LIENS IMAGES



Page de démonstration

Lorsqu'un lien est composé uniquement d'une image, c'est l'alternative de cette image qui constitue son intitulé.

Par exemple, pour les images , c'est le contenu de l'attribut alt qui est l'intitulé du lien :

Dans ces cas, l'alternative de l'image indique à l'utilisateur la destination du lien.

Alternative en fonction du type d'image

Selon le type d'image, l'intitulé du lien (l'alternative de l'image), est défini différemment :

- balise ou balise <area> possédant un attribut href, l'alternative est le contenu de l'attribut
- balise <object>, l'alternative est contenue entre <object> et </object> ;
- balise <canvas>, l'alternative est contenue entre <canvas> et </canvas> ;
- balise <embed>, l'alternative est contenue entre <embed> et </embed>;
- balise <svg>, l'alternative est contenue dans les attributs `title`, `aria-label` ou la balise <desc>.

Liens composites : image et texte

Lorsque le lien est constitué d'une image et d'un texte, si le texte visible est suffisant à comprendre la destination du lien, alors l'image est une image de décoration et doit avoir un alt vide.

LIENS VIDES

Tous les liens doivent avoir un intitulé entre la balise <a> et . Peu importe la mise en forme par la suite, il faut obligatoirement un texte explicite entre ces balises.

Il est courant de trouver des liens vides dépourvus de texte. C'est souvent le cas des liens mis en forme par des icônes générées en CSS ou des images en propriété de fond, mais également souvent le cas d'images HTML (img) dépourvues d'alternative.

Dans le cas des liens mis en forme en CSS avec des icônes, la réparation consiste à mettre un intitulé texte pertinent positionné hors écran. Le graphisme est conservé et les utilisateurs de lecteurs d'écran accèdent à l'information textuelle.

TITRE DE LIENS title

L'attribut title sert surtout à apporter une information complémentaire à l'intitulé du lien et à le rendre explicite. Il doit toujours être construit sur le modèle : intitulé du lien + informations complémentaires.

Les cas de title non conforme sont :

- les title identiques aux intitulés de lien ;
- 👝 les title vides;
- 🦰 les title qui ne reprennent pas au moins le contenu de l'intitulé du lien.

Les seuls cas où l'utilisation de title identiques aux intitulés de lien est acceptable sont les cas des liensimages ou mis en forme par une icône en CSS. Dans ces cas, pour permettre à tous d'accéder à l'information (une icône n'est pas nécessairement comprise par tous, et la destination d'un lien-image n'est pas évidente pour tout le monde), un title identique est autorisé, voire encouragé.



Important :

La présence seule d'un title sur un lien n'en fait pas un lien conforme. Un lien doit toujours avoir un intitulé texte entre les balises <a> et

VOIR AILLEURS / RESSOURCES

• Fiche « Créer du lien sans perdre son chemin » du guide « Contribuer sur le web de manière accessible »

CRITÈRES RGAA

- 6.1 [A]
- 6.2 [A]
- 6.3 [AAA]
- 6.4 [A]
- 6.5 [A]

Fiche 6. Formulaires

INTRODUCTION - CAS UTILISATEURS

L'identification des champs de formulaire est un élément essentiel. Beaucoup d'utilisateurs handicapés vont accéder aux champs de manières très diverses.

Les utilisateurs de lecteurs d'écran disposent de raccourcis clavier leur permettant de naviguer rapidement d'un champ à l'autre et certains dispositifs de navigation vocale proposent d'accéder aux champs par leur nom.

Dans ce type de contexte, il est important que chaque champ de formulaire possède une étiquette liée, afin qu'elle soit restituée lors de la prise de focus. Cela permettra aux personnes aveugles d'utiliser à profit les raccourcis clavier et aux utilisateurs de navigation vocale d'accéder rapidement aux champs. Pour les utilisateurs de dispositifs de pointage adapté, l'étiquette d'un champ permet d'étendre la surface de clic et ainsi améliore l'efficacité des manipulations.

SYNTHÈSE

- Définissez des étiquettes pertinentes.
- Utilisez une méthode conforme pour relier les champs à leurs étiquettes respectives.
- Pour les champs obligatoires avec un format de saisie, pensez toujours à l'indiquer (de préférence dans l'étiquette du champ).
- Pour les erreurs de saisie, reliez l'erreur au champ concerné et indiquez un exemple de saisie réelle lorsque nécessaire.

ÉTIQUETTES

Une étiquette de champ est un texte qui permet d'expliquer quelle donnée est attendue par le champ (nom, prénom, date de naissance, etc.).

Pour être conforme, une étiquette doit :

- être pertinente;
- être correctement reliée au champ correspondant.

Pertinence des étiquettes de champs

Une étiquette est pertinente si elle permet de comprendre la fonction du champ auquel elle est reliée. Il n'est pas forcément nécessaire d'écrire de grandes phrases, mais bien de rester concis. Il faut que l'utilisateur comprenne ce qu'on attend de sa saisie. Par exemple, pour les données d'identité, un label « Nom », un label « Prénom » et un label « Téléphone » sont des étiquettes pertinentes qui permettent de comprendre ce que l'on doit y inscrire.

Pertinence des intitulés de boutons

Les boutons input type="submit", input type="reset" ou encore button doivent avoir des intitulés pertinents qui permettent à l'utilisateur de comprendre l'action du bouton. Ils sont essentiels surtout pour les utilisateurs aveugles afin qu'ils soient sûrs de l'action qu'ils réalisent. Par exemple des intitulés comme « envoyer », « ok », « valider »... ne seront pas suffisamment pertinents. À l'inverse, des intitulés pertinents explicitent l'action, par exemple : « M'inscrire au cours de danse », « Réserver mes billets en ligne ».

Pour rendre les intitulés de boutons pertinents, vous pouvez, soit rendre le texte contenu dans l'attribut value explicite, soit ajouter un title sur l'élément pour l'expliciter.

```
<input type="submit" value="Envoyer le message" />
<input type="submit" value="GO" title="Réserver les billets sélectionnés" />
```

Étiquettes correctement reliées

Plusieurs moyens en HTML et reconnus par le RGAA sont à votre disposition pour labelliser un champ de formulaire.

Page de démonstration de la construction des étiquettes de formulaire

for / id

La méthode la plus courante est la relation HTML entre le champ et son étiquette :

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

L'avantage de cette implémentation par rapport à toutes les autres est qu'elle étend la zone de clic. En effet, lorsque l'utilisateur clique sur le label, le focus est donné directement au champ correspondant. Dans le cas d'une case à cocher, cliquer sur le label permet de cocher la case à cocher.

Attention aux labels implicites : Rien ne vous empêche de les utiliser, néanmoins, les seules méthodes reconnues de labellisations conformes sont celles décrites ici.

Exemple de label implicite avec une relation correcte :

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

Title

Un champ peut également être labellisé via l'attribut title.

```
<input type="text" id="recherche" title="Saisir un mot clé à rechercher"/>
```

Cette implémentation est conforme, mais nous vous invitons à n'utiliser cette implémentation que dans des cas restreints. En effet, l'attribut title n'est visible qu'au survol de la souris. Un utilisateur voyant qui navigue exclusivement au clavier n'accédera pas à cette information. Dans le cas d'un

formulaire complexe, ce type de labellisation peut être un frein à la compréhension pour certains utilisateurs.

L'attribut <u>title</u> devrait être utilisé de préférence dans les cas où le contexte du champ de formulaire permet d'en comprendre la fonction, par exemple, le champ de recherche du site. Il est alors accompagné d'un bouton de validation « Rechercher » ou d'un bouton image en forme de loupe, qui permet de déduire la fonction du champ.

ARIA

ARIA va également permettre de réaliser des étiquettes de formulaires conformes et de réparer certaines implémentations.

```
Votre Nom
<input type="text" aria-labelledby="name" />
```

```
Votre Nom
<input type="text" aria-label="Votre nom" />
```

Attention à la priorité de restitution (calcul du nom accessible) : aria-labelledby et aria-label viennent écraser n'importe quelle étiquette réalisée avec la balise label qui serait déjà implémentée sur un champ. Par exemple :

```
<label for="jfname">Votre nom de jeune fille</label>
<input id="jfname" type="text" aria-labelledby="name" />
Votre nom
```

lci, lors d'un parcours au lecteur d'écran, seule l'étiquette «Votre nom» sera restituée, l'étiquette «Votre nom de jeune fille» sera ignorée.



Important :

Tout comme l'attribut title n'est pas accessible à un utilisateur clavier, les implémentations arialabelledby et aria-label ne sont accessibles qu'aux utilisateurs aveugles. Ainsi, afin que tous les utilisateurs aient le même niveau d'information, ces implémentations doivent toujours être accompagnées d'un texte visible cohérent avec l'étiquette référencée par ces attributs.

placeholder, attention.

placeholder n'est pas considéré comme une méthode de labellisation valide au regard du RGAA, si elle est implémentée seule. La première raison est son contraste insuffisant.

Ensuite, si on prend en compte l'<u>ordre de restitution des éléments définis par la spécification HTML5</u>, l'attribut placeholder est restitué avant l'attribut title. De fait, si des informations importantes sont placées dans l'attribut title qu'on ne retrouve pas dans l'attribut placeholder, alors le champ est mal étiqueté puisque l'utilisateur perd de l'information.

Si vous souhaitez malgré tout utiliser placeholder, assurez-vous soit qu'il n'y a pas d'attribut title sur le champ correspondant, soit que le contenu de l'attribut title est le même que celui du placeholder.

CHAMPS OBLIGATOIRES

Le RGAA n'impose pas que les champs obligatoires soient systématiquement indiqués à l'utilisateur, par exemple il est inutile de le faire sur le seul champ d'un moteur de recherche. Par contre, si vous indiquez les champs obligatoires, cela doit être fait de manière accessible.



Page de démonstration des indications de champs obligatoires

Indication textuelle dans l'étiquette

Selon votre choix d'implémentation d'étiquette de champ (label, title, aria-label...), l'indication de champ obligatoire peut être mise directement dans cette étiquette, par exemple : « Nom (champ obligatoire) ».

Indication par un symbole

Il est très courant de trouver l'indication de champs obligatoire au moyen d'un astérisque. Cette pratique est tout à fait conforme si vous donnez une légende à ce symbole, et ce avant le formulaire concerné.

Indication par la couleur

Si vous décidez de placer les labels des champs obligatoires en rouge pour signifier les champs obligatoires, assurez-vous que cette couleur soit toujours doublée d'un symbole clairement explicité. En effet, un utilisateur aveugle n'a pas accès aux couleurs et un utilisateur ne distinguant pas les couleurs ne pourra pas les percevoir.

```
required et aria-required
```

En HTML5, il est maintenant possible d'utiliser l'API de contrainte de validation, qui permet notamment de définir les champs obligatoires et de gérer un contrôle côté client géré par le navigateur, avec l'attribut required. Cet attribut a aussi la particularité d'être restitué à un utilisateur de lecteur d'écran.

Rien ne vous oblige à utiliser cet attribut, mais si vous l'utilisez, vous devez toujours fournir une alternative visible pour tous les autres utilisateurs. La recommandation est la même si vous utilisez aria-required. Si vous mentionnez les champs obligatoires à l'aide de la propriété aria-label, vous devez également fournir une alternative visible à tous.

AIDES À LA SAISIE

Pour tous les champs obligatoires qui attendent un format particulier, vous devez l'indiquer à l'utilisateur.

Par exemple, un champ de date attend que les valeurs jour, mois et année soient distinguées par une barre oblique pour être conforme. Vous devez donc indiquer « jj/mm/aaaa » qui correspond au format de saisie.

Pour l'indiquer, vous avez à votre disposition les méthodes citées précédemment :

- dans la balise label;
- dans l'attribut title;
- dans le passage de texte relié via l'attribut aria-labelledby.

Vous disposez également de l'attribut aria-describedby qui vous permet de référencer la valeur de l'identifiant d'un passage de texte contenant l'indication de format. L'utilisation de aria-describedby sur le champ de formulaire aura pour effet de restituer le passage de texte en complément du label quelle que soit la méthode qui aura été employée : <label>, title, etc. contrairement à aria-labelledby.

```
<label for="date">Jour de naissance</label>
<input type="text" id="date" aria-describedby="format-date" />
jj/mm/aaaa
```

ERREURS DE SAISIE

Lorsque le formulaire retourne des erreurs, l'erreur doit être mentionnée soit :

- dans l'étiquette du champ (<label>, title, aria-label, aria-labelledby);
- dans un passage de texte avant le formulaire, qui rassemble la liste de tous les champs erronés ;
- dans un passage de texte relié au champ de saisie avec l'attribut aria-describedby.

Il vous est également possible d'indiquer ces erreurs avec la propriété aria-label et la propriété aria-label invalid. Dans ces cas, vous devez toujours fournir une alternative visible pour tous les autres utilisateurs.

```
<label for="date">Jour de naissance - Erreur - Veuillez renseigner ce champ
obligatoire</label>
<input type="text" id="date" aria-invalid="true" aria-describedby="format-
date" />

p id="format-date">jj/mm/aaaa
```

Au niveau AA, les messages d'erreur de saisie relative à l'utilisation d'un mauvais format de données, par exemple une erreur de saisie de courriel, doivent comporter un exemple de saisie réelle.

♀ Astuce :

aria-describedby et aria-labelledby peuvent recevoir plusieurs valeurs d'identifiants séparées par des espaces. L'ordre de déclaration correspond à l'ordre de restitution par un lecteur d'écran.

LISTES DE CHOIX

Les listes de choix <select> doivent être, comme tout champ de formulaire, labellisées correctement.

Si la liste propose un choix limité de réponses, alors rien n'est demandé, mis à part que les options soient présentées dans un ordre cohérent (par exemple par ordre alphabétique).

```
<label for="civilite">Civilité</label>
  <select id="civilite">
      <option>Madame</option>
      <option>Monsieur</option>
  </select>
```

Par contre, si la liste de choix propose un certain nombre d'options pour lesquelles il est possible de faire des regroupements, alors vous devez utiliser la balise de regroupement optgroup pourvue d'un attribut label qui titre ce regroupement.

REGROUPEMENT DE CHAMPS

Dans certains formulaires, il est nécessaire de créer des regroupements de champs. Ces regroupements de champs sont utiles pour ne pas induire certains utilisateurs, notamment aveugles, en erreur. Par exemple, un formulaire d'achat en ligne propose un formulaire avec la saisie d'une adresse de livraison et une adresse de facturation. Puisque les deux vont posséder des champs avec des étiquettes similaires, il est impératif de créer un regroupement pour que l'utilisateur comprenne quelle adresse il renseigne. Les regroupements de champs doivent posséder eux aussi une étiquette pertinente dans la balise legend.

```
<fieldset>
  <legend>Adresse de facturation</legend>
  <label for="nomf">Nom</label>
  <input type="text" id="nomf" />
  <label for="ruef">Rue</label>
  <input type="text" id="ruef" />
  <label for="villef">Ville</label>
  <input type="text" id="villef" />
  </fieldset>

  <fieldset>
  <legend>Adresse de livraison</legend>
  <label for="noml">Nom</label>
  <input type="text" id="noml" />
  <label for="ruel">Rue</label>
  <input type="text" id="ruel" />
  <label for="villel">Ville</label>
  <input type="text" id="ruel" />
  <label for="villel">Ville</label>
  <input type="text" id="villel" />
  </fieldset>
```

Les autres cas typiques d'application sont les groupes de cases à cocher ou de boutons radio.

VOIR AILLEURS / RESSOURCES

• Calcul du nom accessible des éléments de formulaires (ressource en anglais)

CRITÈRES RGAA

• 11.1 [A]

• 11.2 [A]

• 11.3 [AA]

• 11.4 [A]

• 11.5 [A]

• 11.6 [A]

• 11.7 [A]

• 11.8 [A]

• 11.9 [A]

• 11.10 [A]

• 11.11 [AA]

• 11.14 [AAA]

• 11.15 [AAA]

Fiche 7. La prise de focus et la navigation clavier

INTRODUCTION - CAS UTILISATEURS

La prise de focus est l'état renvoyé par un élément qui reçoit l'attention suite à une action de l'utilisateur. Il y a trois moyens en HTML de donner le focus à un élément :

- en activant l'élément par un dispositif de pointage (souris);
- en activant l'élément par la touche tabulation ;
- en activant l'élément par un raccourci clavier.

La prise de focus concerne particulièrement les utilisateurs naviguant exclusivement au clavier. Sont concernés les utilisateurs qui ont des problèmes de motricité et qui ne peuvent pas utiliser une souris efficacement. Ils vont donc utiliser principalement le clavier pour se déplacer sur la page. Sont concernés également, les utilisateurs dépourvus de motricité comme les tétraplégiques et qui ne peuvent utiliser que des outils externes pour utiliser un ordinateur. Ils vont utiliser des systèmes de pointages particuliers : vue, licorne (headstick), etc.

En CSS, la prise de focus est contrôlable par le pseudo-évènement : focus , l'aspect de l'indication visuelle du focus par la propriété outline.

SYNTHÈSE

- Ne définissez jamais de valeurs pour les propriétés de l'outline qui dégraderaient sa visibilité.
- Pour les liens indiqués uniquement par la couleur, définissez une mise en forme autre que la couleur qui permet d'identifier le survol de la souris et la prise de focus.
- Naviguez sur vos pages à la tabulation et assurez-vous que la prise de focus est cohérente et que vous ne restez pas bloqué sur un élément (piège au clavier).

CONSERVER L'OUTLINE

Le RGAA impose de conserver la propriété outline. Dans les feuilles de style, il est interdit de trouver des propriétés visant à supprimer cet outline à la prise de focus (:focus), notamment :

outline: 0
outline: none
outline: transparent

De même, les propriétés suivantes ne doivent pas être dégradées, c'est-à-dire que les valeurs définies ne doivent pas aboutir à une perte de visibilité :

- outline-color : par exemple outline-color: transparent est considéré comme non conforme;
- outline-width: outline-width: 0 est considéré comme non conforme;
- outline-style : outline-style:hidden est considéré comme non conforme.

Même si vous utilisez une bordure (border) pour signifier la prise de focus, cette alternative n'est pas considérée comme pertinente.

En effet, outline est une propriété gérée par le navigateur. Certains navigateurs proposent des mécanismes qui permettent d'augmenter la visibilité de cet outline. Ainsi, si dans vos feuilles de style, vous indiquez outline:0, les paramètres du navigateur visant à augmenter l'outline seront invisibles.

Remarque : désactiver la propriété outline pour les états :hover et :active n'est pas considéré comme non conforme. Seules les propriétés attachées à l'état :focus d'un élément sont prises en compte ici.

Visibilité du survol et la prise de focus pour les liens

Lorsque les liens en environnement de texte sont signalés uniquement par la couleur (la couleur différente est suffisamment contrastée par rapport au texte environnant), notamment parce que la propriété native text-decoration des liens a été annulée (text-decoration:none), alors une indication visuelle autre que la couleur permet de signifier la prise de focus et le survol de la souris.

Cette indication visuelle peut être l'apparition d'un soulignement, la mise en gras du texte du lien, l'insertion d'une icône.

Ordre de Tabulation Cohérent

L'ordre de tabulation est l'ordre dans lequel le focus se déplace (vers un élément suivant ou vers un élément précédent). L'ordre naturel est l'ordre de succession des éléments dans le code source. La tabulation va se déplacer successivement entre tous les éléments interactifs.

Une mise en forme en grille qui ferait que le focus parcourt d'abord la colonne visuellement au centre puis la colonne de droite n'est pas nécessairement considérée comme un ordre incohérent, si dans le code source, les éléments se succèdent de manière logique et compréhensible. Un ordre cohérent n'est pas nécessairement une prise de focus de gauche à droite.

Un ordre de tabulation incohérent peut être la conséquence d'une redéfinition des index de tabulation via l'attribut tabindex. L'attribut tabindex peut recevoir des valeurs numériques, et si ces valeurs sont supérieures à 0, alors l'ordre n'est plus l'ordre naturel, mais celui redéfini par tabindex. Cette modification de l'ordre de tabulation peut également être la conséquence d'un script (JavaScript) qui redéfinit les prises de focus.

Par exemple:

```
<a href="#" tabindex="3">Lien 1</a>
<a href="#" tabindex="1">Lien 2</a>
```

La tabulation ira d'abord sur le « Lien 2 », puis le « Lien 1 ».

Il est fortement déconseillé d'utiliser des valeurs différentes de 0 ou -1 pour tabindex. tabindex="0" permet à un élément qui ne reçoit pas naturellement le focus d'entrer dans le parcours de la tabulation.

Par exemple, les éléments qui reçoivent naturellement le focus sont les liens <a>, boutons <button>, champs de formulaires <input />, etc. À l'opposé, une image < img /> ne reçoit jamais le focus. Si, sur cette image, nous implémentons tabindex="0", alors l'image entre dans le parcours du focus.

À l'inverse, tabindex="-1" permet de retirer un élément du parcours du focus. L'élément qui possède cet attribut ne recevra plus le focus.

```
<a href="#" tabindex="-1">Lien</a>
```



Important :

Contrôlez **toujours** les éléments pour lesquels vous définissez tabindex="-1". Vérifiez bien son utilisation dans vos pages et les problèmes de restitution ou d'accès que cela pourrait engendrer.

Ordre de tabulation et textes cachés

Un cas courant d'ordre de tabulation non cohérent se rencontre dans l'utilisation de composants interactifs qui permettent d'afficher ou masquer des éléments.

En effet, vous devez vous assurer que les textes cachés sont correctement restitués. Une façon de s'en assurer est de faire se succéder logiquement dans le code source l'élément qui permet de déclencher l'apparition d'un contenu caché et le contenu caché. Si, en plus, ce contenu caché possède des éléments interactifs (formulaires par exemple), alors l'ordre de tabulation sera cohérent.

Lorsque le texte caché est contrôlé par un motif de conception ARIA, alors l'ordre de tabulation est défini par le motif de conception. Il doit être scrupuleusement respecté.

PIÈGE AU CLAVIER

Lorsque le parcours de tous les éléments interactifs de la page, à l'aide de la seule touche de tabulation, est impossible, alors on se trouve devant un piège au clavier.

Guide de l'intégrateur RGAA 3

Exemples de piège au clavier :

- Lorsque le focus quitte un lien, un script lance le rechargement de la page. L'utilisateur qui navigue au clavier est condamné à ne jamais pouvoir dépasser ce lien (et atteindre l'élément suivant focusable) puisque le fait de quitter le lien recharge la page et renvoie le focus en haut de la page.
- Un formulaire capture le focus et empêche le focus de quitter le champ tant qu'une date n'a pas été sélectionnée. Malheureusement, la saisie d'une date n'est possible qu'en cliquant sur un calendrier. De fait, l'utilisateur qui navigue au clavier ne peut pas sortir de ce champ pour aller sélectionner une date, il est condamné à rester bloquer à ce niveau et ne pourra jamais aller au-delà.

Il existe d'autres cas, mais ceci devrait vous permettre de les identifier facilement.

La meilleure solution est de vous assurer de l'absence de ces pièges au clavier et de trouver un moyen de les éviter tant que possible. Toutefois, si ce piège au clavier est inévitable, compte tenu d'une conception particulière (ou d'une insertion d'un script tiers sur lequel vous n'avez pas la main), vous devez mettre un moyen à disposition de l'utilisateur, et l'en informer, lui permettant d'accéder au clavier à l'élément suivant ou précédent pouvant recevoir le focus. Ce peut être par exemple, un lien d'évitement qui permet de ne pas prendre le focus sur cet élément et ainsi éviter le piège.

- 10.7 [A]
- 10.13 [A]
- 12.13 [A]
- 12.14 [A]

Fiche 8. Respecter la distinction fond et forme

INTRODUCTION - CAS UTILISATEURS

Pour créer un site web, on utilise principalement deux langages : HTML et CSS. HTML permet de créer un balisage sémantique de votre contenu, c'est le « fond ». CSS permet de réaliser des mises en forme (couleurs, positionnements, etc.), c'est la « forme ». Les deux ont des objectifs différents et ne doivent jamais jouer un rôle différent. Simplement : la mise en forme ne doit jamais apporter du sens au contenu et le contenu ne doit pas être pourvu d'éléments de mises en forme. Votre site doit être compréhensible sans la mise en forme.

Les utilisateurs aveugles qui naviguent avec un lecteur d'écran n'ont pas accès à la mise en forme. Donc si vous implémentez une image avec du contenu en propriété de fond (background-image), ce contenu est totalement inexistant pour ces utilisateurs.

Certains utilisateurs ayant des problèmes de vues divers (vision des couleurs, taille des caractères, police particulière) vont charger une feuille de style utilisateur ou utiliser des fonctionnalités dédiées pour adapter la présentation de votre site à leur propre besoin de mise en forme. Si vous aviez défini des éléments qui apportent de l'information (comme des images de fond, des contenus générés en CSS), ces éléments vont disparaître et ces utilisateurs n'y auront pas accès.

SYNTHÈSE

Désactivez les styles de vos pages et :

- comparez la même page avec les styles activés et assurez-vous qu'aucun élément de contenu n'a disparu ;
- assurez-vous que les contenus restent compréhensibles et que leur succession dans le code source est cohérente.

UTILISER LES BALISES HTML POUR LEUR SENS PROPRE

Vous ne devez pas utiliser les balises à des fins de présentation. En effet, l'utilisation de balises HTML pour une finalité différente de celle prévue dans les spécifications (au regard du type de document déclaré) est interdite.

Par exemple:

- 👝 utilisation des balises <hx> à seule fin de créer un effet typographique ;
- dutilisation de la balise <blockquote> à seule fin de mettre un paragraphe en retrait (puisque cette balise possède nativement un style qui créée une indentation).



Important:

L'utilisation d'éléments <div> ou pour créer des paragraphes est considérée comme non conforme. Vous devez toujours utiliser l'élément lorsque vous rédigez ou insérez un paragraphe.

MISE EN FORME EXCLUSIVE VIA CSS

Toutes vos mises en forme doivent être réalisées uniquement en CSS. CSS met à disposition 3 moyens pour réaliser les mises en forme :

- la liaison de la page HTML avec une feuille de style externe link rel="stylesheet" href="#" media="screen">;
- la balise <style></style> dans le <head> de la page;
- l'attribut style sur des éléments HTML.

Il est interdit de trouver dans vos pages des éléments et attributs de mise en forme. En effet, les feuilles de style peuvent être surchargées par une feuille de style utilisateurs, mais il est beaucoup plus compliqué et quelquefois impossible de surcharger les mises en forme générées via des éléments ou attributs.

Tous les attributs de mise en forme sont interdits (align, alink, background, basefont, bgcolor, border, color, link, text, vlink,...).

width et height sont autorisés uniquement sur les éléments , <svg>, <object>, <embed> et <canvas>.

Les éléments <basefont>, <blink>, <center>, , <marquee>, <s>, <strike>, <tt> et <u> sont interdits.

CONTENU VISIBLE SANS CSS

Images de fond

Les images de fond (insérées par la propriété CSS background ou background-image) ne doivent jamais être porteuses d'information.

Lorsque vous en êtes à l'étape de conception de vos gabarits, face à la maquette graphique, prenez soin de distinguer les éléments qui seront des images, et dans ces images, de distinguer celles qui sont porteuses d'information et celles qui sont purement décoratives.

Lorsque les images sont purement décoratives (couleur de fond, illustration décorative), vos choix d'intégration sont libres, à condition de respecter les règles d'accessibilité :

- Les images de décoration sont implémentées en propriétés de fond CSS, il n'y a a priori aucun problème d'accessibilité.
- Les images de décoration seront implémentées en image HTML par exemple, veillez donc à définir une alternative vide .

Les images porteuses d'information sont très souvent des images contenant du texte en image. Dans ces cas, il ne faut pas en faire des images de fond, mais les implémenter en image HTML en veillant à définir une alternative pertinente.

Nous renvoyons à la fiche sur les images pour les différents types d'implémentation d'images, notamment au travers des balises <svg>, <object>, <embed> et <canvas>.

Lorsqu'aucune autre solution n'est possible, il est toléré de transmettre l'information véhiculée par l'image de fond via un texte masqué de manière accessible (en positionnement par exemple). Néanmoins, cette solution peut poser d'autres problèmes lorsque sont utilisés des dispositifs spécifiques, comme certains modes de contrastes élevés.

Contenu généré en CSS content

La propriété CSS content permet d'injecter du contenu dans le HTML. Pour autant, même si ce contenu est visible avec les CSS activés, il ne l'est plus lorsque les CSS sont désactivées. Ainsi, vous ne devez donc jamais utiliser content pour insérer du contenu texte.

content et les pseudos-sélecteurs :after et :before sont très souvent utilisés pour l'insertion d'icônes grâce à des polices d'icônes. Si les icônes sont purement décoratives, alors il n'y a pas de problème d'accessibilité. Par contre, s'il s'agit d'icônes porteuses d'information, alors vous devez veiller à ce qu'un moyen alternatif permette de récupérer cette information.

L'exemple emblématique (mais pas unique) de cette pratique est la création de liens ou de boutons avec des icônes. L'implémentation est très souvent lacunaire puisque ces liens et boutons sont au final des éléments vides.





```
<a href="#"><span class="fa fa-home"></span></a>
<button><span class="fa fa-cog"></span></button>

.fa-home:before {
  content: "\f015";
}
.fa-cog:before {
  content: "\f013";
}
```

Dans ces cas, un utilisateur aveugle qui n'a pas accès à la mise en forme ne sait pas à quoi sert le lien ou le bouton.

Vous avez des contraintes graphiques et vous ne pouvez mettre un texte visible à la place de ces icônes. Une solution acceptable est de fournir un texte positionné hors écran. De plus, ces icônes ne sont pas forcément compréhensibles par tout le monde, il est donc généralement préférable (mais non obligatoire) d'ajouter un attribut title qui reprend ce texte positionné hors écran, pour que tous les utilisateurs aient la même information.

A

Important :

La présence seule d'un title sur un lien n'en fait pas un lien conforme. Un lien doit toujours avoir un intitulé texte entre les balises <a> et

D'autres solutions sont à votre disposition :

- utiliser des images HTML avec une alternative alt renseignée;
- utiliser des images <svg> avec un élément <desc> renseigné.

Couleur de fond pour les textes

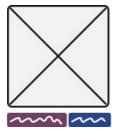
Pour chaque couleur de police de texte (propriété CSS color) déclarée dans vos styles, cette police doit avoir une couleur de fond déclarée (propriété CSS background-color), au moins héritée d'un parent. Littéralement, il suffit d'avoir une couleur de fond sur l'élément body (parent de n'importe quel enfant du document). Pour aider l'utilisateur, il est préférable de prendre ce critère de manière plus stricte. Le cas le plus courant est de trouver une couleur de fond d'élément simulée par une image de fond. Lorsque les images sont désactivées (cas d'utilisateurs présentant des troubles de l'attention, par exemple), cette

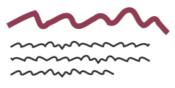
couleur n'est plus disponible et le texte n'est plus lisible.

CONTENU COMPRÉHENSIBLE SANS CSS

C'est souvent à cause de contraintes de mise en forme que la structure HTML est altérée au détriment de la compréhension.

Par exemple, votre maquette propose une mise en forme des actualités avec une image-lien et les thèmes de l'actualité sur la gauche et sur la droite, le titre et le chapeau de l'actualité.





Votre première réaction est de placer dans le code source : d'abord l'image, puis les mots clés, puis le titre et enfin le chapeau.

```
<div class="left">
  <a href="#"><img src="..." alt="lien" /></a>

    thème 1
    thème 2
    </div>
</div>
<div class="right">
    <h2>Titre de l'actualité</h2>
    Chapeau de l'actualité
</div></div>
```

Pourtant le titre est également le titre du lien et des thèmes. Un ordre cohérent voudrait que le titre **précède** tous les éléments auxquels il se rapporte dans le code source. Libre à vous de réaliser ensuite les mises en forme que vous souhaitez.

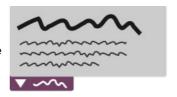
De plus, si plusieurs actualités se succèdent, comment un utilisateur, notamment aveugle qui navigue au lecteur d'écran, va-t-il être en mesure de comprendre quel titre se rapporte à quel lien et à quels thèmes ? Si l'utilisateur suit un ordre logique linéaire de consultation, sa compréhension sera fausse parce que la succession des éléments dans le code source ne reflète pas le sens, compréhensible uniquement par la mise en forme.

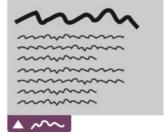
La construction correcte dans ce cas est donc :

Il faut toujours garder à l'esprit que la mise en forme (la maquette du site) ne doit pas être fondée sur des choix de structure qui répondent uniquement à des demandes de mises en forme. Lorsque vous modifiez ou choisissez une certaine implémentation des éléments HTML, et un certain ordre de succession des éléments, posez-vous toujours la question de savoir si dans cet ordre, sans la mise en forme, il est possible de comprendre les informations sans équivoque.

Textes cachés correctement positionnés

Lorsqu'un élément est rendu visible sur action de l'utilisateur, le composant qui a permis de rendre visible l'élément caché doit être situé avant dans le code source. La raison simple est qu'un utilisateur aveugle qui parcourt la page de manière linéaire n'aura pas conscience de l'apparition du texte caché, si le composant est situé après.





Dans cet exemple, un bouton permet d'étendre la zone de contenu pour afficher une partie de contenu qui était cachée. Lorsque l'on active le bouton, le focus reste dessus. Une fois le contenu apparu, le focus est toujours sur ce bouton. L'utilisateur aveugle va alors parcourir le contenu suivant en pensant accéder au contenu qu'il a affiché avec le bouton, ce qui n'est pas le cas dans cet exemple.



Exception:

Lorsque le texte caché appartient à un composant ARIA, l'ordre dans lequel il doit être implémenté, par rapport à l'évènement déclencheur, est imposé par le motif de conception ARIA utilisé.

VOIR AILLEURS / RESSOURCES

- Script de gestion du chargement des polices d'icônes et mise en place d'alternatives accessibles (Filament Group - AfontGarde - ressource en anglais)
- Stylish Extension Firefox permettant de gérer facilement des feuilles de style utilisateurs
- Banque de feuilles de style utilisateurs

- 8.9 [A]
- 10.1 [A]
- 10.2 [A]

- 10.3 [A]
- 10.13 [A]

Fiche 9. Images

INTRODUCTION - CAS UTILISATEURS

Les images véhiculent parfois une information non textuelle. Cette information, qui peut aider à comprendre le contenu auquel elle se rapporte, doit être accessible à tous.

Fournir une alternative est indispensable pour les utilisateurs qui ne perçoivent pas le contenu visuel. Un lecteur d'écran ou une loupe vocalisée vont pouvoir accéder à cette alternative et la restituer à l'utilisateur ou bien l'information sera affichée si l'utilisateur désactive les images.

SYNTHÈSE

- Définir un alt sur toutes les images .
- Définir un alt="" sur les images de décoration.
- Définir une alternative conforme et pertinente en fonction du type d'image lorsqu'elle est porteuse d'information.
- Éviter au maximum les images textes.

ALTERNATIVE OBLIGATOIRE

Toutes les images doivent avoir un attribut alt. Cette obligation tient au fait qu'en l'absence de cet attribut, un lecteur d'écran restitue le chemin ou le nom du fichier source, ce qui n'a pas de sens pour l'utilisateur.

Les éléments concernés sont :

- les images ;
- les zones réactives <area>;
- les champs de formulaire <input type="image">.

IMAGE DE DÉCORATION : ALTERNATIVE VIDE

Si l'image ne véhicule aucune information, l'image n'a pas vocation à être restituée. Son alternative doit alors être vide. De plus, elle ne doit pas posséder d'attribut title.

```
<img src="url.png" alt="" />
```

Nous vous renvoyons à la <u>fiche « Quand une image vaut mille mots »</u> du guide « Contribuer sur le web de manière accessible » pour de plus amples informations sur les différences entre les images de décoration et les images porteuses d'information.

Cas des images svg

Dans le cas des images vectorielles, vous devez ajouter la propriété aria-hidden="true" et vous assurez qu'aucun attribut ou aucune balise ne sont présents afin de labelliser l'image :

- les balises <title> et <desc> sont absentes ou vides ;
- la balise <svg>, ou l'un de ses enfants, est dépourvue d'attribut title ;
- la balise <svg>, ou l'un de ses enfants, est dépourvue de rôle, propriété ou état ARIA visant à labelliser l'image vectorielle (aria-label, aria-describedby, aria-labelledby par exemple).

En résumé, une image vectorielle de décoration devrait ressembler à ceci :

```
<svg role="img">
</svg>
```

Cas des images <object>, <canvas> et <embed>

Dans le cas des images
ou <canvas>, vous devez ajouter la propriété aria-hidden="true" et
vous devez vous assurer qu'aucun attribut ou aucune balise ne sont présents afin de labelliser l'image :

- l'alternative textuelle entre les balises ouvrantes et fermantes (<object> et </object> , ou <canvas> et </canvas>) est vide ;
- la balise (<object> ou <canvas>), ou l'un des ses enfants, est dépourvue de role, propriété ou état ARIA visant à labelliser l'image (aria-label, aria-describedby, aria-labelledby par exemple).

Concernant la balise <embed>, elle doit posséder un attribut aria-hidden="true" et elle, ou l'un des ses enfants, doit être dépourvue de rôle, propriété ou état ARIA visant à labelliser l'image (aria-label, aria-describedby, aria-labelledby par exemple).

IMAGE PORTEUSE D'INFORMATION: ALTERNATIVE RENSEIGNÉE

Une image porteuse d'information apporte une information par elle-même ou essentielle à la compréhension du contenu auquel elle est associée. L'alternative doit être courte et suffisante pour que les utilisateurs qui ne peuvent pas voir l'image comprennent l'information véhiculée.

Pour les images (img, input type="image", area), vous devez renseigner l'attribut alt.

S'il est présent, le contenu de l'attribut title est égal au contenu de l'attribut alt.

Images vectorielles

Pour les images vectorielles, vous devez :

- ajouter un role="img" sur l'élément svg;
- ajouter une propriété aria-label contenant l'alternative de l'image;
- ajouter une balise <desc> qui contient un passage de texte identique à la propriété aria-label.



Attention, s'il est présent, l'attribut title doit être équivalent à aria-label et <desc>.

Si ces implémentations ne sont pas possibles, vous pouvez créer un lien adjacent à l'image qui mène vers une alternative pertinente et identique au contenu de la balise <desc>.

```
Images <object type="image">, <canvas> et <embed type="image">
```

Pour ce type d'images, vous pouvez soit labelliser l'image directement, soit mettre en place une alternative.

Si vous souhaitez mettre en place une alternative, vous devez :

- faire suivre l'image d'un lien adjacent permettant d'afficher une page ou un passage de texte contenant une alternative pertinente;
- proposer un mécanisme qui permet à l'utilisateur de remplacer l'image par un texte alternatif pertinent;
- ou proposer un mécanisme qui permet à l'utilisateur de remplacer l'image par une image possédant une alternative pertinente.

Note : dans le cas des images <canvas>, l'alternative peut être donnée par un texte présent entre les balises <canvas> et </canvas>.

Si vous souhaitez labelliser l'image :

- ajouter la propriété aria-label qui contient le texte permettant de donner l'alternative pertinente de l'image;
- ou ajouter la propriété aria-labelledby qui référence un passage de texte pertinent.

Attention, s'il est présent, l'attribut title doit être équivalent à aria-label ou aria-labelledby selon l'implémentation choisie.

Compatibilité avec les technologies d'assistance

Pour les implémentations d'images avec les éléments <svg> et <canvas>, assurez-vous de la restitution des alternatives avec les lecteurs d'écran et navigateurs de la base de référence. En effet, les implémentations, notamment ARIA, étant très changeantes dans les navigateurs et lecteurs d'écran, l'important, au-delà de respecter le balisage, est de vous assurer de leur restitution par un lecteur d'écran.

IMAGES TEXTES

Lorsque des images textes peuvent être reproduites en HTML et CSS, vous ne devez pas utiliser d'image. Ceci est tout à fait possible :

- lorsque les polices utilisées ne sont pas des polices complexes,
- lorsque l'image ne joue pas sur une déformation de la police,
- ou encore, lorsque la police n'est pas une police qu'il serait compliqué à reproduire.

Dans tous les autres cas, les textes doivent être réalisés en HTML et CSS, afin de permettre aux utilisateurs qui en ont besoin d'adapter leur lisibilité (agrandissement des caractères, modification des couleurs, des polices...).

Il est également possible de conserver les images textes et de fournir à l'utilisateur un mécanisme de remplacement qui lui permet, avec un bouton par exemple, de remplacer toutes les images textes en textes stylés.

CAPTCHA

Dans le cas des images tests (Captcha), il faut renseigner l'alternative de manière à permettre à l'utilisateur de comprendre la nature et le rôle de l'image.

Par exemple, alt="saisir le code présent dans l'image".

Ce mécanisme devra être complété par une méthode alternative permettant à l'utilisateur de récupérer le code pour pouvoir poursuivre.

Cette problématique sera détaillée dans un futur guide destiné aux développeurs.

IMAGE LÉGENDÉE

HTML5 introduit les balises figure et figcaption afin de pouvoir associer une légende à une image.

Il est important de le faire lorsque la légende contient des informations sur l'image, par exemple un copyright, qu'un utilisateur aveugle pourrait mal interpréter en pensant que l'information de la légende concerne le texte associé.

L'utilisation de figure et figcaption réclame certaines adaptions pour combler le manque de support

éventuel par les technologies d'assistance.

```
<figure role="group">
  <img alt="illustration" src="img.png"/>
  <figcaption>
  Illustration - copyright 2016 Simon Jérémi
  </figcaption>
  </figure>
```

Le texte « illustration » dans l'alternative de l'image crée une liaison sémantique entre l'image et sa légende, dans le cas où l'élément figure ne serait pas restitué par les lecteurs d'écran. Le texte « illustration » est repris dans la légende pour établir la liaison sémantique.

Page de démonstration des images légendées

VOIR AILLEURS / RESSOURCES

• Fiche « Quand une image vaut mille mots » du guide « Contribuer sur le web de manière accessible »

- 1.1 [A]
- 1.2 [A]
- 1.3 [A]
- 1.4 [A]
- 1.5 [A]
- 1.8 [AA]
- 1.9 [AAA]
- 1.10 [A]

Fiche 10. Informations par la forme et la couleur

INTRODUCTION - CAS UTILISATEURS

Les informations données par la forme et les informations données par la couleur impactent de nombreux utilisateurs. En premier lieu, les personnes aveugles qui ne peuvent pas voir les couleurs ou les formes, mais également les utilisateurs qui ne voient pas ou ne distinguent pas certaines couleurs ou combinaisons de couleurs. Pour tous ces utilisateurs, une information donnée uniquement par la forme ou la couleur sera ignorée.

SYNTHÈSE

- Ne pas donner d'informations uniquement par la couleur.
- Ne pas donner d'informations uniquement par la forme.
- Implémenter des alternatives à la couleur et à la forme qui soient pertinentes.

INFORMATIONS PAR LA COULEUR

Le cas le plus courant d'information par la couleur est l'indication de la page active dans le menu de navigation. Dans ce cas, une réparation simple consiste à fournir l'information sous forme textuelle afin qu'un utilisateur qui ne perçoit pas les couleurs (aveugle ou utilisateur ne percevant pas les contrastes) puisse accéder à la même information. Dans le cas du lien actif, on pourra tout simplement ajouter dans le titre du lien une mention sur le modèle : « Accueil - page active ».

Le principe à garder en tête est que toute couleur qui véhicule une information doit être accessible par un autre moyen, notamment un élément textuel.



Important :

À noter que l'utilisation d'une information donnée par la forme comme alternative à une information donnée par la couleur sera insuffisante, les utilisateurs aveugles ne voyant ni forme ni couleur.

INFORMATIONS PAR LA FORME, TAILLE OU POSITION

Les informations données par la forme, la taille ou la position regroupent une grande variété de cas. Il peut s'agir d'indiquer par un effet CSS qu'un contenu est « affiché » ou « masqué », d'indiquer dans un nuage de tags le score de pertinence par la taille des caractères, d'indiquer l'élément actif d'un contenu par un effet CSS comme un triangle inséré en propriété de fond.

Dans ces cas, la réparation la plus simple est pratiquement identique à celle pour les informations données par la couleur : mettre à disposition un équivalent soit avec une image avec une alternative (alt="") pertinente, soit à l'aide d'une icône doublée d'un texte alternatif positionné hors écran.

Dans l'extrait de code suivant, la période sélectionnée indiquée par un triangle noir utilise une image (balise) avec une alternative pertinente.



Important :

À noter que l'utilisation d'une information donnée par la couleur comme seule alternative à une information donnée par la forme serait insuffisante. Les utilisateurs aveugles ne voyant ni forme ni couleur.

VOIR AILLEURS / RESSOURCES

• La section « Information par la couleur » de la fiche « Couleurs » dans le guide « Contribuer sur le Web de manière accessible »

- 3.1 [A]
- 3.2 [A]
- 10.14 [A]
- 10.15 [A]

Fiche 11. Agrandissement des caractères



Page de démonstration

INTRODUCTION - CAS UTILISATEURS

La bonne adaptation des contenus d'une page web lorsque la taille de la police est augmentée est primordiale pour les utilisateurs malvoyants. Ces utilisateurs vont vouloir agrandies uniquement la taille de la police, et non pas la page (zoom graphique). La mise en forme doit alors s'adapter convenablement pour que tous les textes restent lisibles et compréhensibles à certains niveaux de zoom.

SYNTHÈSE

- Il ne faut pas définir des tailles de police en pixels.
- Il faut éviter de fixer des hauteurs de boîte en pixels.
- Il faut utiliser la propriété overflow: hidden avec précaution.
- Il est préférable de définir des points de rupture en em.

AGRANDISSEMENTS DES CARACTÈRES EN FONCTION DU NAVIGATEUR

L'agrandissement des caractères (taille de la police) est à différencier du zoom graphique. Dans de nombreux navigateurs, le zoom (Ctrl +) agrandit l'ensemble de la page. Vous pouvez vous en rendre compte, car même les images sont agrandies.



Pour augmenter uniquement la taille de la police, selon le navigateur :

 Dans Firefox, dans « Affichage » aller à « Zoom » et sélectionner « Zoom texte seulement ». À partir de ce moment, Ctrl + agrandit uniquement les tailles de polices.

- Dans Safari, dans « Présentation » cochez « Réduire/agrandir le texte seulement ». À partir de ce moment, Ctrl + agrandit uniquement les tailles de polices.
- Dans Internet Explorer, dans « Affichage » (ou « Page » selon la version) aller à « Taille du texte » et sélectionner une des valeurs proposées (par exemple « La plus grande »).

Toutes les recommandations et explications données ici sur les tailles de polices ne concernent que les feuilles de style et les déclarations à destination des écrans (media="screen" ou media="all").

TAILLES DE POLICES EN UNITÉS RELATIVES EXCLUSIVEMENT

La première des choses est de s'assurer que toutes vos tailles de polices déclarées le soient en unités relatives.

Vous pouvez utiliser les unités suivantes :

- em, rem;
- %;
- vw, vh (unités relatives à la taille du viewport);
- xsmall, small, large... (mots clés).

A contrario, les unités fixes pour définir des tailles de police sont interdites. Parmi les tailles fixes, on trouve par exemple :

- px (pixel);
- pt (point);
- cm (centimètres).

Cette obligation tient du fait que dans certains navigateurs (notamment Internet Explorer), si les tailles de polices sont définies en unités fixes, les paramètres d'agrandissement de la taille des caractères restent sans effet.

LISIBILITÉ LORSQUE LA TAILLE DE LA POLICE EST AGRANDIE.

Au niveau AA du RGAA, le texte doit rester lisible et on ne doit pas perdre d'informations lorsque la taille de la police par défaut est augmentée de 200%.

200% correspond à 6 fois l'action Ctrl + sur Firefox (avec l'option « Texte seul » du zoom). À noter : le contrôle de cette lisibilité ne peut être fait efficacement que sur Firefox ou Safari, Internet Explorer ne disposant pas de l'option zoom texte seul.

Attention aux tailles de boîtes fixées

Lorsque vous réalisez vos mises en forme, vous êtes souvent tentés de fixer les hauteurs ou les largeurs pour avoir un rendu harmonieux pour certains éléments.

Lorsque les tailles de boîtes sont fixées par les propriétés CSS : height, width, max-height et max-width dont les valeurs sont définies en pixels, elles ne peuvent s'agrandirent proportionnellement à la

Guide de l'intégrateur RGAA 3

taille de la police. Le contenu qui débordera ne sera plus totalement lisible, voire illisible.

Attention toutefois, cela ne concerne que les boîtes qui contiennent ou sont susceptibles de contenir du texte.

Vous pouvez utiliser le pixel pour les dimensions et les positionnements des boîtes qui ne contiennent pas de texte ou ne sont pas susceptibles de contenir ou de servir de fond à du texte.

De préférence, remplacez dans vos feuilles de style :

- width par min-width;
- height par min-height.

À noter que width et height ne posent aucun problème lorsque les valeurs sont définies en em.

Évitez également au maximum d'utiliser les propriétés max-height et max-width sans en contrôler le contenu et le comportement à l'agrandissement des caractères.

Autres propriétés CSS à utiliser avec précaution

overflow:hidden

Pour contrôler des débordements et pour coller à un graphisme défini, vous pouvez être tentés d'utiliser la propriété overflow:hidden. Si la boîte en question ne contient pas de texte, vous pouvez à peu près être assurés que cela ne posera pas de problème. Par contre, si cette boîte contient du texte et qu'en plus vous avez défini une hauteur fixée en pixel, le texte disparaîtra.

Les positionnements

Attention aux positionnements qui ne suivent pas le flux standard, notamment les positionnements absolute et fixed. Assurez-vous que l'agrandissement des caractères ne provoque pas de chevauchement avec ces boîtes positionnées hors du flux.

Gérer les espaces et les retours à la ligne

La propriété white-space: nowrap empêche le contenu texte d'être soumis au retour à la ligne automatique. Cette propriété doit donc être utilisée avec précaution. Effectuez toujours des tests pour évaluer le comportement à l'agrandissement des caractères.

CONVERTIR SES TAILLES DE POLICES DU PIXEL AU EM

On dit qu'une unité est relative par rapport à l'élément parent. Par exemple, vous avez un élément div qui possède une taille de police de 1.5em. Cet élément div possède un élément p qui possède une taille de police de 0.9em. Alors la taille calculée de votre élément p, et donc celle appliquée est de 1.35em c'est-à-dire 0.9em x 1.5em.

RESPONSIVE ET AGRANDISSEMENT DES CARACTÈRES : UNE PIERRE, DEUX COUPS

Aujourd'hui, la plupart des sites web sont dits responsive, ils s'adaptent aux différentes tailles d'écrans. Cela se fait très majoritairement par l'utilisation des media-queries en CSS.

Les points de rupture déclarés dans ces media-queries dépendent très fortement des différentes tailles d'écrans sur le marché, elles sont donc très souvent déclarées en pixels.

Mais vous pourriez très bien les déclarer en em, et ainsi provoquer le réarrangement des éléments que vous avez définis pour les tailles d'écrans, à la taille de la police. Et d'une pierre deux coups ! En effet, le fonctionnement est simple : plus vous augmentez la taille de police par défaut (Ctrl +), plus la taille disponible en em diminue.

Par exemple:

```
@media screen and (min-width:767px) {
}
```

pourrait être remplacé par

```
@media screen and (min-width:47.9375em) {
   /** 767px / 16 = 47.9375em **/
}
```

16px correspond à la taille par défaut définie par le navigateur.

Ce dispositif est excellent pour adapter vos contenus aux capacités réelles des tailles d'écrans de vos utilisateurs.

VOIR AILLEURS / RESSOURCES

• Article du W3C sur les unités en CSS

- Critère 10.4 [AA]
- Critère 10.10 [AAA]

Fiche 12. Multimédia

INTRODUCTION - CAS UTILISATEURS

La majeure partie des contenus multimédias que vous rencontrerez sont des vidéos (avec ou sans bande-son) ou des fichiers sons.

Entrent également dans la catégorie de contenus multimédias, les éléments interactifs comme les animations ou publicités en Flash par exemple.

Tous les utilisateurs sont impactés par cette thématique puisque tous les sens sont sollicités dans la plupart des vidéos : la vue pour voir les images et les actions à l'écran, l'ouïe pour entendre les paroles et la motricité pour arrêter ou relancer la vidéo.

La grande majorité des critères de la thématique « Multimédia » du RGAA 3 se concentrent sur les alternatives mises à disposition des utilisateurs (sous-titres, transcription textuelle, audiodescription...). Ces alternatives sont du ressort du concepteur, du contributeur, mais pas de l'intégrateur. Pour cette raison, ces problématiques ne sont pas traitées ici, mais pouvez en retrouver une partie sur la <u>« Fiche 4 : Du multimédia pour tous et toutes »</u> du guide « Contribuer accessible sur le web de manière accessible ».

D'autres critères concernent les composants interactifs permettant de contrôler les contenus multimédias (les boutons de lecture, pause, volume, etc.). Ces problématiques seront traitées dans un futur guide à destination des développeurs.

SYNTHÈSE

- Faire précéder toutes les vidéos d'un titre.
- Dans le cas des contenus multimédias non temporels, fournir un lien clairement identifié permettant d'accéder à l'alternative accessible.

IDENTIFIER LA VIDÉO

Lorsque vous insérez du contenu multimédia, vous devez l'identifier, le mieux étant de le faire précéder d'un titre. Ce titre peut reprendre celui de l'élément lui-même s'il en possède un. Il doit permettre de comprendre quelle est l'information présentée dans ce contenu. Tout élément multimédia doit être considéré comme un contenu à part entière et donc structuré par un titre.

Nous vous renvoyons à la fiche <u>« Les titres, ou le plan de la page »</u> du guide « Contribuer accessible sur le web de manière accessible » pour la mise en place de ce titre.

CONTENUS MULTIMÉDIA NON TEMPORELS

Les animations de type Flash sont des contenus multimédia non temporels (une carte interactive en Flash par exemple). Pour ce type d'éléments, il est demandé à ce qu'une alternative accessible soit fournie.

En tant qu'intégrateur, votre travail est de fournir le moyen d'accès à l'alternative de manière pertinente. Pour cela, depuis la page qui possède le contenu multimédia non temporel, vous devez fournir un lien, clairement identifiable et à proximité du contenu, qui mène vers l'alternative accessible. L'alternative peut être contenue soit dans la page elle-même (le lien sera alors une ancre), soit dans une autre page.

VOIR AILLEURS / RESSOURCES

• <u>« Fiche 4 : Du multimédia pour tous et toutes »</u> du guide « Contribuer accessible sur le web de manière accessible ».

- Critère 4.15 [A]
- Critère 4.16 [A](et