

BTS SIO

-

DOCUMENT DE STAGE

-

GUIDE D'ACCESSIBILITE

Sommaire :

INTRODUCTION :	5
Contexte	5
Sources	5
Accessibilité numérique	5
Obligations légales et sanctions	6
Schéma pluriannuel de mise en accessibilité	6
Déclaration d'accessibilité	7
ARIA :	8
Les attributs	8
Rôles	8
Aria-*	9
IMAGES :	10
Critères et enjeux	10
Les images informatives	10
Mise en place de l'alternative	10
Les images décoratives	11
Vérifications	11
Les images avec légende / texte	11
Les images avec légende	11
Les images qui contiennent du texte	12
Les images CAPTCHA	12
CADRES :	13
Le titre du cadre	13
Le contenu	13
Les cadres techniques	13
COULEURS :	14
L'information donnée par la couleur	14
Les contrastes	14
CONTENUS MULTIMEDIA :	16
Les médias non temporels	16
Les médias temporels	16
Le Son	17

Les Sous-titres _____	17
La transcription textuelle _____	18
L'audiodescription _____	19
Contrôle et compatibilité _____	19
TABLEAUX : _____	20
Tableaux de données simples et complexes _____	20
Tableaux de mise en forme _____	22
LIENS : _____	23
Intitulé visible et nom accessible _____	23
Contexte du lien _____	24
Visibilité du lien _____	24
SCRIPTS : _____	25
Gabarits de conception ARIA (Design Pattern) _____	25
Interfaces sans gabarit _____	25
Base de référence _____	26
Compatibilité au clavier (et aux dispositifs de pointage) _____	26
Messages de statut & Zone Live _____	27
Changement de contexte _____	27
ÉLEMENTS OBLIGATOIRES : _____	28
Respect des standards _____	28
Code de la langue et Sens de lecture: _____	28
Titre de page _____	29
STRUCTURATION DE L'INFORMATION : _____	30
Hiérarchiser les contenus _____	30
Structure du document _____	30
Structure des listes _____	31
Citations _____	32
PRÉSENTATION DE L'INFORMATION : _____	33
Mise en forme et contenus _____	33
Mise en forme _____	33
Contenus _____	33
Agrandissement du texte _____	34
Contexte _____	34
Contraintes _____	34
Conseils _____	34

Couleurs et fonds	35
Focus et contenus cachés	35
Informations et contrôles	35
FORMULAIRES :	37
Étiquette de champ	37
Regroupement de champs	38
Boutons	38
Contrôle de saisie	39
Données	40
NAVIGATION :	41
Moyens de navigation	41
Lien d'évitement	41
Regroupement et tabulation	42
Contenus additionnels	42
CONSULTATION :	43
Limite de temps	43
Nouvelles fenêtres	43
Téléchargements	43
Contenu Cryptique	44
Gestes complexes	44
Annuler une action	44
Orientation de l'écran	45
Changements et mouvements	45

INTRODUCTION :

Contexte

Le RGAA (Référentiel Général d'Amélioration de l'Accessibilité) est destiné à définir les différentes modalités d'accessibilité du contenu HTML des pages web.

La version actuelle est la 4.1 publiée en 2021.

Le RGAA est aussi conforme aux 50 critères de la norme internationale WGAC (*Web Content Accessibility Guidelines*) 2.1 retenus dans la norme européenne.

Le RGAA contient quant à lui 106 critères différents portant sur toutes les caractéristiques d'une page web (que ce soit au niveau visuel mais aussi au niveau pratique). Ces critères devront bien évidemment être testés.

Sources

Support de la formation accessibilité des applications (DSIN)

[Référentiel général d'amélioration de l'accessibilité - RGAA](#)

[Accessibilité numérique - DesignGouv](#)

[Référentiel général d'amélioration de l'accessibilité — Wikipédia](#)

Accessibilité numérique

L'accessibilité numérique consiste à **rendre les services de communication au public en ligne accessibles aux personnes handicapées**, il faudra ainsi rendre les pages web :

- **Perceptibles** : Faciliter la perception visuelle et auditive du contenu par l'utilisateur, proposer des équivalents textuels au contenu non textuel, créer un contenu pouvant être présenté de différentes manières avec les mêmes informations...
- **Utilisables** : Fournir à l'utilisateur des éléments d'orientation pour trouver le contenu, rendre toutes les fonctionnalités accessibles au clavier, laisser à l'utilisateur suffisamment de temps pour lire le contenu...
- **Compréhensibles** : Aider l'utilisateur à corriger des erreurs de saisie, permettre une compréhension rapide des fonctionnalités...
- **Robustes** : Optimiser la compatibilité avec les utilisateurs, y compris avec les technologies d'assistance

Le handicap est défini comme : Toute limitation d'activité ou restriction de participation à la vie en société subie dans son environnement par une personne

L'accessibilité numérique s'effectue sous **tous types d'handicap** :

- Handicap **physique** : Visuel, sensoriel...
- Handicap **mental** : Psychologique, cognitif, psychique...

Obligations légales et sanctions

Depuis 2012, tous les sites publics appartenant aux services de l'État (ou collectivités territoriales) ont une obligation d'accessibilité.

Depuis 2019, tous les sites publics doivent publier une déclaration d'accessibilité et afficher leur conformité sur la page d'accueil.

L'obligation d'accessibilité ne s'effectue pas uniquement sur les sites accessibles à tous, elle s'étend aux **sites intranet, extranet** ainsi qu'aux **application mobiles** et **progiciels**.

Il faudra ainsi sur toutes les pages **afficher le taux de conformité** :

"Accessibilité : totalement | partiellement | non conforme" (100% | >50% | <50% de conformité)

Il faudra **produire et publier un schéma pluriannuel de mise en accessibilité** sur 3 ans

Tout manquement peut entraîner une **sanction de 20 000€** par service en ligne.

Schéma pluriannuel de mise en accessibilité

Le schéma pluriannuel de mise en accessibilité est un schéma permettant de montrer la mise en accessibilité du site web, il contiendra de nombreuses informations comme :

- Les différentes missions du référent accessibilité numérique
- Les ressources (humaines & financières) affectées à l'accessibilité numérique
- Les actions de formation / sensibilisation des agents

...

(voir ce site pour tout ce qu'il contient : [Schéma pluriannuel de mise en accessibilité - DesignGouv](#))

Le schéma pluriannuel de mise en accessibilité devra être fourni dans la déclaration d'accessibilité !

Déclaration d'accessibilité

Lorsque tous les tests sont effectués, il faut **déclarer l'accessibilité du site web**, pour cela il faudra suivre ces différentes étapes :

- Déclaration de la mise en place de la mise en conformité
- État de conformité du site
- Résultats des tests
- Contenus non accessibles
- Information sur la déclaration d'accessibilité

(Pour plus de détails, consultez ce site : [Exemple de déclaration d'accessibilité - RGAA](#))

La déclaration d'accessibilité sera valide à partir de sa date de publication jusqu'à modification ou refonte du site sinon **pendant 3 ans** (ou 18 mois si nouvelle version). Il sera tout de même préférable de mettre régulièrement la déclaration à jour avec les évolutions sur site web.

ARIA :

Principe d'ARIA : Améliorer l'accessibilité des contenus (HTML / JavaScript...) par l'ajout de la **sémantique** et **d'informations structurales & comportementales**.

Les attributs

Rôles

L'attribut Role fournit diverses informations sur le type de rôle d'un élément HTML. Il existe [différents types de rôles](#) Il y a plus de 70 rôles différents, pouvant être répartis par groupe.

- Les rôles doivent être attribués pour **tous les éléments principaux** de la page web (Les **Landmark**) qui permettent de regrouper des contenus.

→ Entête du site `<header role="banner">`

→ Navigation `<nav role="navigation">`

→ Région principale `<main role="main">`

→ Bas de page `<footer role="contentinfo">`

...

- Les rôles doivent aussi être attribués pour **les éléments interactifs** de la page web (Les **Widget**)

→ Boutons : `role="button"`

→ Liens : `role="link"`

→ Barres de progression : `role="progressbar"`

...

- Les rôles sont aussi attribués pour **les éléments structurant la page web** :

→ Titres : `role="title"`

→ Images : `role="img"`

→ Listes : `role="list" ou role="listitem"`

→ Tableaux : `role="table" (et plus particulièrement role="row" ou role="cell")`

...

- Les rôles peuvent être attribués à **des actions** ou à **des fenêtres** :

- Alertes : *role="alert"*
- Infos / logs : *role="log"*
- Fenêtres d'alertes : *role="alertdialog"*
- Fenêtres modales : *role="dialog"*

Aria-*

Aria-* est aussi un ensemble de rôles / attributs complétant les éléments HTML afin d'être utilisé par les différents outils d'assistance (dont les widgets JavaScript). Aria-* comporte environ 40 rôles différents.

- Aria-* peut être attribué pour déterminer **un état**, ou **une propriété** :

- Élément développé : *aria-expanded*
- Élément caché / affiché : *aria-hidden*
- Élément (dé)sélectionné : *aria-selected*
- Élément (in)actif : *aria-disabled*
- Nom d'un élément : *aria-label*
- Niveau hiérarchique : *aria-level*

...

- Aria-* peut aussi être attribué pour des **éléments live** (éléments pouvant changer) :

- Éléments priorisés : *aria-live* (polite : messages à priorité peu élevée (informe dès que possible) / assertive : message à priorité très élevée (avertit directement l'utilisateur)).
- Éléments modifiés : *aria-atomic* (false : uniquement la partie modifiée / true : l'ensemble de ce qui a été modifié)
- Éléments en modification : *aria-busy*

- Aria-* peut aussi être attribué pour les **relations entre les différents éléments** :

- Relation parent - enfant : *aria-owns*
- Élément qui identifie / décrit l'élément actuel : *aria-labelledby* / *aria-describedby*
- Élément étant un message d'erreur : *aria-errormessage* & *aria-invalid*

IMAGES :

Critères et enjeux

Les images sont un enjeu majeur, notamment pour **amener directement de l'information** ou pour **décorer une page web**, mais pour les malvoyants ou aveugles, il faut pouvoir y apporter une alternative.

Le **RGAA** a défini de nombreux **critères pour les images**, en voici certains :

- Chaque image informative a-t-elle une alternative textuelle ?
- Chaque image décorative est-elle bien ignorée par les technologies d'assistance ?
- Chaque légende d'image est-elle bien reliée à l'image correspondante ?

...

Il faut aussi **prendre en compte le contexte** (éléments hors image), et ainsi **ne pas répéter les mêmes informations** si elles sont déjà citées précédemment.

Les images informatives

Le **RGAA** a imposé deux critères pour les images informatives :

- Chaque image porteuse d'information a-t-elle une alternative textuelle ?
- Pour chaque image porteuse d'information ayant une alternative textuelle, cette alternative est-elle pertinente ?

Il faut donc **prendre en compte le contexte** pour pouvoir vérifier ces deux critères. Ainsi si l'information véhiculée par **l'image est nécessaire à la compréhension**, il faut absolument **fournir une alternative**.

Il faut malgré tout fournir une alternative à la fois **courte** et **claire sans détailler l'ensemble des informations** de l'image.

Mise en place de l'alternative

Afin de mettre en place une alternative, il faut ajouter **l'attribut "alt"** dans la balise img :

Ex : alt="Exemple de description"

Il est aussi possible d'utiliser l'**attribut "title"** en complément si besoin.

Pour les **balises svg** contenant l'attribut `role="img"`, on pourra fournir une alternative avec **une balise <title>** ou avec **les attributs aria-label ou aria-labelledby**.

Les images décoratives

Contrairement aux images informatives, les images décoratives **ne comportent aucune information au contenu**, selon le RGAA, elles doivent répondre à **un critère** :

→ Chaque image de décoration est-elle correctement ignorée par les technologies d'assistance ?

Vérifications

Pour veiller à ce que l'assistance technique ne lise pas les images décoratives, il faut dans la balise `` avoir une **alternative vide : alt=""** et **ne pas avoir d'attribut title**.

Pour les balises svg, il faut qu'elles **possèdent l'attribut `aria-hidden="true"`**, de plus il faut vérifier que ces balises **ne contiennent pas de <title>, <desc> ou d'attribut `aria-label`, `aria-labelledby`**...

Les images avec légende / texte

Les images avec légende

Les images avec légende doivent **vérifier un critère clé pour le RGAA** :

→ Chaque légende d'image est-elle, si nécessaire, correctement reliée à l'image correspondante ?

La légende est associée à l'image, mais **il ne faut pas confondre alternative** (vu précédemment pour les images informatives) **et légende**. **L'alternative** contient les **informations présentes dans l'image** tandis que **la légende** contient des **informations supplémentaires à l'image** (auteur, date, contexte...).

En HTML5, il faudra **utiliser les balises <figure> et <figcaption>** afin de lier l'image et sa légende. De plus il faudra **utiliser les attributs `role="figure"` et `aria-label="légende"`**.

Les images qui contiennent du texte

Comme pour les images avec légende, les images qui contiennent du texte doivent aussi **vérifier un critère du RGAA** :

→ Chaque image texte porteuse d'information, en l'absence d'un mécanisme de remplacement, doit si possible être remplacée par du texte stylé.

Ainsi, une image texte **devra si possible être remplacée par du contenu HTML stylé en CSS** afin de permettre à certains utilisateurs (dont les déficients visuels) qui ont du mal à lire les textes en image de **les modifier ou les agrandir** afin de les voir plus facilement.

Il y a tout de même **quelques exceptions** comme **les logos** (ou dénominations commerciales) ainsi que **les images CAPTCHA** ou **les images tests**.

Les images CAPTCHA

Une image CAPTCHA est une image de vérification essentielle pour de nombreux sites afin de **vérifier que l'utilisateur n'est pas un robot**.

L'image CAPTCHA est principalement une (ou plusieurs) images. L'utilisateur doit ainsi retrouver quelque chose de précis (voiture, montagne, bateau...), il pourra bien sûr y avoir certaines dérivées.

Les images CAPTCHA sont problématiques pour les déficients visuels puisque les images **sont souvent difficiles à interpréter**, ainsi le RGAA a **imposé deux critères** :

→ Pour chaque image utilisée comme CAPTCHA ou comme image test, ayant une alternative textuelle, cette alternative permet-elle d'identifier la nature et la fonction de l'image ?

→ Pour chaque image utilisée comme CAPTCHA, une solution d'accès alternatif au contenu ou à la fonction du CAPTCHA est-elle présente ?

Ainsi il faudra ajouter un texte alternatif au CAPTCHA ainsi que si possible, associer l'image à un lecteur audio afin de permettre aux utilisateurs de remplir le CAPTCHA même sans pouvoir percevoir les images.

CADRES :

Les cadres (ou iframes) peuvent contenir de nombreux contenus plus ou moins riches, **tous les utilisateurs doivent donc savoir ce qu'ils contiennent**. Pour cela, **deux critères sont imposés par le RGAA** pour les cadres :

- Chaque cadre a-t-il un titre de cadre ?
- Ce titre de cadre est-il pertinent ?

Le titre du cadre

Le titre du cadre devra **informer à la fois de la nature du document** (site web, vidéo...), mais aussi de **son contenu** (contenu du site web, de l'article, de la vidéo...).

Pour cela il faudra **utiliser l'attribut *title="nature : exemple de contenu"* dans la balise `<iframe>`**.

Le contenu

Si les informations contenues *dans le cadre* sont rédigées, financées, développées, ou sous le contrôle d'un organisme, elles devront **être accessibles**.

Les autres informations non rédigées, financées, développées devront **respecter les règles d'accessibilité au mieux** sans obligation d'accessibilité.

Les cadres techniques

Certains **cadres ne comportant pas de contenu à restituer** aux technologies d'assistance **devront être cachés**, ce sont les "*cadres techniques*".

Afin de les cacher, il faudra **utiliser l'attribut *aria-hidden="true"*, l'attribut *tabindex="-1"* afin d'éviter le focus sur le cadre** ainsi qu'un *title="contenu technique"* si l'élément est quand même lu par l'outil d'assistance.

COULEURS :

La perception des couleurs est une réelle problématique dans la compréhension des sites web et applications, en effet **certaines personnes ne peuvent pas ou perçoivent différemment les couleurs** (déficients visuels, daltoniens...).

Certaines images ou informations peuvent véhiculer l'information en utilisant la couleur, ainsi il faudra **vérifier que les informations ne soient pas uniquement véhiculées par les couleurs**, pour cela, **le RGAA impose 3 critères pour les couleurs** :

- Dans chaque page web, l'information ne doit pas être donnée uniquement par la couleur.
- Dans chaque page web, le contraste entre la couleur du texte et la couleur de son arrière-plan est-elle suffisamment élevée ?
- Dans chaque page web, les couleurs utilisées dans les composants d'interface ou les éléments graphiques porteurs d'information sont-elles suffisamment contrastées ?

L'information donnée par la couleur

Pour les différentes informations données par la couleur (*ex : fait / à faire en vert / rouge*), il faut **prévoir une alternative afin de permettre une compréhension des informations sans la couleur** :

Exemple : Ajouter un texte "Fait" avec un fond vert et "À faire" avec un fond rouge.

Dans cet exemple il n'y a pas obligatoirement besoin de la couleur pour comprendre quelles lignes sont faites ou à faire.

Pour les champs obligatoires dans les formulaires, il faudra veiller à ne pas uniquement les mettre en rouge, il faut ainsi **ajouter "(obligatoire)" et l'attribut required** afin que l'utilisateur soit informé de cette obligation. De plus il faudra **ajouter l'attribut `aria-hidden="true"` dans la balise "(obligatoire)"** afin de ne pas avoir de redondance de lecture pour les technologies d'assistance.

Les contrastes

Pour ce qui est du contraste entre un texte et son arrière-plan, il faut séparer texte en gras et sans gras.

Texte en gras :

- Texte inférieur à 18.5px : contraste d'**au moins 4,5:1**
- Texte supérieur (ou =) à 18;5px : contraste d'**au moins 3:1**

Texte normal :

→ Texte inférieur à 24px : contraste d'**au moins 4,5:1**

→ Texte supérieur (ou =) à 24px : contraste d'**au moins 3:1**

Pour ce qui est des composants d'interface (liens, boutons, menus...) ainsi que des **éléments graphiques porteurs d'informations** (images graphiques / statistiques...), le contraste est **d'au moins 3:1**.

[Pour pouvoir tester ces différents contrastes](#)

CONTENUS MULTIMEDIA :

Les contenus multimédias doivent être **impérativement accessibles à tous** puisqu'ils constituent **énormément d'informations**, il existe deux types de médias :

→ Les **Médias temporels** : vidéos avec ou sans son, fichiers audios

→ Les **Médias non-temporels** : Animations / publicités (GIF, SVG, canvas...).

Les médias sont de plus en plus présents, il faut ainsi **préparer des solutions alternatives pour qu'ils soient accessibles à tous**, pour les utilisateurs avec une déficience à la fois **visuelle** (audiodescription, transcription textuelle) et **auditive** (sous-titres, langue des signes...).

Le RGAA impose 13 critères différents pour les médias qui seront énoncés au fur et à mesure.

Les médias non temporels

Les médias non temporels constituent une **part infime des médias** puisqu'ils sont actuellement peu voire plus utilisés. **Le RGAA a cependant imposé des critères** si un média non temporel est utilisé :

→ Chaque média non temporel a-t-il, si nécessaire, une alternative ? Si oui, est-elle pertinente ?

Pour ce qui est des **images (GIF)**, il faudra utiliser **une alternative : alt="..."**, pour ce qui est des animations flash ou contenu incorporé il faudra utiliser une **balise de texte <p> pour les animations flash dans l'object** ou pour le **contenu incorporé dans une balise <embed> (ou <noembed>)**.

Les médias temporels

Il est indispensable d'identifier un média temporel afin d'éviter à l'utilisateur de les parcourir pour les comprendre, pour cela **le RGAA a imposé un critère** :

→ Chaque média temporel est-il clairement identifiable ?

Afin de les identifier, il faut **ajouter un titre** en prévoyant dans les développements **un champ titre ainsi qu'un choix de niveau de titre <h...>**.

Le Son

Certaines balises (ou codes JavaScript) peuvent déclencher automatiquement un son, ce qui est peu utilisé (trop intrusif) mais lors de son utilisation il faut vérifier un critère du RGAA :

→ Chaque son déclenché automatiquement est-il contrôlable par l'utilisateur ?

Les sons sont donc à éviter, si c'est impossible il faut veiller à ce que l'utilisateur puisse le contrôler (son et/ou lecture).

Les Sous-titres

Afin de **permettre à tous de comprendre un média temporel**, il faut **trouver une alternative au son**, pour cela, **l'utilisation de sous-titres** est nécessaire, ces sous-titres répondent à **des critères du RGAA** :

→ Chaque média temporel synchronisé préenregistré a-t-il, si nécessaire, des sous-titres synchronisés ? Si oui sont-ils pertinents ?

Les sous-titres doivent permettre de **comprendre les informations** (dialogues, voix-off, sons...) **simplement en les lisant**, ils doivent être **synchronisés avec la vidéo** (dans sa langue) et doivent être **(dés)activables**. Il ne faut malgré tout pas confondre sous titrage de traduction et sous titrage pour sourds et malentendants.

Le sous titrage automatique de Youtube n'étant pas conforme à l'accessibilité (RGAA) (ils sont peu fiables : mots manquants, mal transcrits, mal synchronisés...) il faut ainsi **créer ses propres sous-titres**.

Il existe plusieurs formats de sous-titres :

→ Le **SRT** (SubRIP Text),

→ Le **Web VTT** (The Web Video Text Tracks Format)

→ Le **TTML** (Timed Text Markup Language).

Le texte de sous titrage devra **respecter plusieurs critères** :

→ Faire 1 ou 2 lignes (codées en UTF-8) avec 35/40 caractères max par ligne

→ Affichage pendant 1 à 10 secondes

→ Possibilité de formater le texte (italique, gras...)

Il est aussi possible de **changer la couleur du texte du sous-titre** afin de leur donner un sens:

- **Blanc** : Locuteur(s) à l'image (#ffffff).
- **Jaune** : Locuteur(s) hors champ / voix off (#ffff00).
- **Cyan** : Pensée(s) / Flash-back (#00ffff).
- **Magenta** : Musique(s) (#ff00ff).
- **Rouge** : Indications sonores / Descriptions / Bruits (#ff0000).
- **Vert** : Langues étrangères (#00cc00).

Rendre accessible les vidéos via les sous-titres est désormais indispensable. Voici [quelques autres conseils pour sous-titrer correctement une vidéo](#).

La transcription textuelle

La transcription textuelle est **structurée en HTML** et est complète (comparé aux sous-titres qui défilent en fonction du média temporel, la transcription textuelle est disponible en intégralité à n'importe quel moment).

Le RGAA **impose des critères pour la transcription textuelle** :

→ Chaque média temporel préenregistré a-t-il, si nécessaire, une transcription textuelle ? Si oui, celle-ci est-elle pertinente ?

Si l'audiodescription est présente sur une vidéo, la transcription textuelle n'est pas obligatoire.

La **transcription textuelle contient la totalité de ce qui est exprimé oralement** dans les médias temporels ainsi que **toutes les informations descriptives** nécessaires à une compréhension équivalente de l'action comme : Les dialogues, les textes, les images / graphiques, le nom des locuteurs / objets, le lieu ainsi que tout ce qui se passe dans la vidéo.

Pour une transcription textuelle propre, il faut veiller à :

- Mentionner la musique de fond,
- Utiliser des majuscules (lettres capitales) pour informer d'un cri,
- Ne pas censurer les insultes / gros mots,
- Ne pas traduire les informations dites dans une autre langue,
- Indiquer lorsque le contenu audio est incompréhensible.

L'audiodescription

Comparé aux sous-titres qui ne fournissent comme information uniquement ce qu'on entend (dialogues, son...), **l'audiodescription d'une vidéo fournit de l'information à propos des personnages, des changements de scènes, du texte apparaissant...** Cependant lorsque l'information est déjà donnée dans la piste audio aucune audiodescription supplémentaire n'est ajoutée. L'audiodescription est soumise à **des critères du RGAA** :

→ Chaque média temporel préenregistré a-t-il, si nécessaire, une audiodescription synchronisée ? Si oui celle-ci est-elle pertinente ?

Il n'y a malheureusement **pas d'intégration native de l'audiodescription aux navigateurs**, ainsi il faudra soit proposer **deux vidéos, une avec et une sans audiodescription** ou mettre en place un **lecteur vidéo capable de gérer l'audiodescription**.

Contrôle et compatibilité

Le RGAA impose deux règles de contrôle et une règle de compatibilité :

→ La consultation de chaque média temporel est-elle, si nécessaire, contrôlable par le clavier et tout dispositif de pointage ?

→ La consultation de chaque média non temporel est-elle contrôlable par le clavier et tout dispositif de pointage ?

→ Chaque média temporel et non temporel est-il compatible avec les technologies d'assistance ?

Il faut **respecter plusieurs points de contrôle** du média :

→ Pouvoir mettre lecture / pause et stop,

→ Gestion et (dés)activation du son,

→ (Dés)Activation des sous-titres s'il en possède,

→ (Dés)Activation de l'audiodescription s'il en possède

Pour ce qui est de la compatibilité, le média **doit absolument être compatible avec les technologies d'assistance, les intitulés / rôles / valeurs... doivent donc être accessibles**.

TABLEAUX :

Il existe **trois types de tableaux différents : simples, complexes, et de mise en forme**. Les tableaux sont faciles à comprendre lorsqu'on les voit, mais pour les déficients visuels il est difficile de parcourir un tableau (navigation séquentielle).

Le RGAA impose plusieurs critères pour l'ensemble des tableaux :

→ Pour chaque tableau de données, chaque en-tête de colonnes et chaque en-tête de ligne sont-ils correctement déclarés ?

→ Pour chaque tableau de données, la technique appropriée permettant d'associer chaque cellule avec ses en-têtes est-elle utilisée ?

Il faut ainsi veiller à ce que les **cellules d'en-têtes soient structurées avec une balise <th>**, de plus il faut que les **cellules d'entête de colonne possèdent l'attribut *scope="col"*** et que les **cellules d'entête de ligne possèdent l'attribut *scope="row"***. Pour finir il faut que ces balises d'entête aient un rôle (*role="rowheader" / role="columnheader"*)

Tableaux de données simples et complexes

Les tableaux de données simples sont des tableaux possédant uniquement des en-têtes réparties sur la 1^{ère} ligne et/ou la 1^{ère} colonne.

Les tableaux de données simples **n'ont pas forcément besoin de titre**, cependant s'il existe il doit absolument être pertinent.

Les tableaux de données complexes sont des tableaux composés d'en-têtes pouvant être réparties sur plusieurs lignes et/ou colonnes (la portée n'est donc pas forcément valable pour l'ensemble de la ligne / colonne).

Les tableaux de données simples possédant un titre et les tableaux de données complexes doivent respecter deux règles du RGAA :

→ Pour chaque tableau de données ayant un titre, le titre est-il correctement associé au tableau de données ?

→ Pour chaque tableau de données ayant un titre, celui-ci est-il pertinent ?

Il faut donc ajouter un **titre clair (pas trop long) et cohérent** au tableau **dans la balise <caption> dans l'attribut *title="..."***, il faut aussi penser à le renseigner dans *l'aria-label* (ou dans un texte associé au tableau via *aria-labelledby*).

Chaque balise <th> d'un tableau complexe doit avoir un id unique et chaque balise <td> doit posséder un attribut headers avec l'id de l'entête associé.

Un résumé est aussi obligatoire pour les tableaux complexes, selon le RGAA :

→ Chaque tableau de données complexe a-t-il un résumé ? Si oui, celui-ci est-il pertinent ?

Il faut donc qu'un **résumé soit intégré dans la balise <caption>** directement après l'ouverture de la balise <table>.

Exemples de tableaux de données simple et complexes :

Tableau Simple :

```
<table>
  <caption>Températures moyennes mensuelles pour l'année 2016</caption>
  <tr>
    <td></td>
    <th scope="col">Paris</th>
    <th scope="col">Marseille</th>
  </tr>
  <tr>
    <th scope="row">Juin</th>
    <td>22°C</td>
    <td>28°C</td>
  </tr>
  <tr>
    <th scope="row">Juillet</th>
    <td>24°C</td>
    <td>32°C</td>
  </tr>
</table>
```

Tableau Complexe :

```
<table>
  <caption>Température moyennes mensuelles sur deux années&nbsp;: le premier
  regroupement concerne l'année 2016, le second l'année 2017. La première colonne
  représente les mois, la seconde la ville de Brest et la troisième la ville de Lyon.
  </caption>
  <tr>
    <th colspan="3" id="2016">2016</th>
  </tr>
  <tr>
    <td></td>
    <th id="brest">Brest</th>
    <th id="lyon">Lyon</th>
  </tr>
  <tr>
    <th id="juin1">Juin</th>
    <td headers="2016 brest juin1">18°C</td>
    <td headers="2016 lyon juin1">26°C</td>
  </tr>
  <tr>
    <th id="juillet1">Juillet</th>
    <td headers="2016 brest juillet1">24°C</td>
    <td headers="2016 lyon juillet1">34°C</td>
  </tr>
  <tr>
    <th colspan="3" id="2017">2017</th>
  </tr>
  <tr>
    <th id="juin2">Juin</th>
    <td headers="2017 brest juin2">19°C</td>
    <td headers="2017 lyon juin2">27°C</td>
  </tr>
  <tr>
    <th id="juillet2">Juillet</th>
    <td headers="2017 brest juillet2">25°C</td>
    <td headers="2017 lyon juillet2">35°C</td>
  </tr>
</table>
```

Tableaux de mise en forme

Les tableaux de mise en forme sont **fortement déconseillés** puisqu'ils sont **difficiles à parcourir**. S'ils sont malgré tout utilisés, ils doivent remplir ces **deux conditions du RGAA**:

→ Pour chaque tableau de mise en forme, le contenu linéarisé reste-t-il compréhensible ?

→ Chaque tableau de mise en forme ne doit pas utiliser d'éléments propres aux tableaux de données. Cette règle est-elle respectée ?

Les tableaux de mise en forme ne doivent **pas utiliser les éléments propres aux tableaux de données** vus précédemment :

→ Pas de titres (balise `<caption>`, `title...`) ni de pied de tableau (`<tfoot>`)

→ Pas d'entête ni de liaison avec les entêtes (`<th>`, `scope`, `headers`, `role="rowheader"...`) ...

Le tableau de mise en forme doit être linéarisé et doit **rester compréhensible**, de plus la **balise `<table>` doit posséder un attribut `role="presentation"`**. Seules les **balises `<table>`, `<tr>` et `<td>` sont donc utilisées**.

LIENS :

Les liens permettent de **naviguer de page en page**, ils sont **définis par 3 éléments** : son nom, sa description et son contexte. Pour les liens **le RGAA n'impose que deux critères** :

→ Chaque lien est-il explicite ?

→ Dans chaque page web, chaque lien, à l'exception des ancres a-t-il un intitulé ?

Les liens doivent être **compréhensibles et accessibles** à la fois pour une lecture rapide mais aussi pour faciliter la compréhension du lien par les déficients visuels et moteurs.

Il ne faut pas confondre lien et bouton, même s'ils sont tous les deux cliquables **un lien permet de naviguer vers une autre page** (balise `<a href>` ou attribut `role="link"`) tandis que **le bouton permet de réaliser une action** (menus, slider, affichage de contenu...) (balise `<button>` ou attribut `role="button"`).

Il faut donc différencier **le lien** qui **emmène quelque part** avec le **bouton** qui **déclenche une action**. Le **lien** sera activé **uniquement avec la touche Entrée** tandis que le **bouton** sera activé par **la touche Entrée et Espace**.

Intitulé visible et nom accessible

Chaque lien doit avoir un intitulé explicite pour identifier son objectif, sa fonction et sa destination, de plus si le lien provoque un changement de contexte (comme l'ouverture d'un nouvel onglet), il faut l'indiquer, selon le RGAA :

→ Chaque lien est-il explicite ?

L'intitulé visible d'un lien est le texte contenu entre les balises `<a>...`, il doit être visible en permanence pour permettre aux outils d'assistance d'identifier rapidement cet intitulé.

Le nom accessible d'un lien est le nom qui sera restitué par les technologies d'assistance, le nom accessible sera alors lu selon l'ordre suivant :

→ Passage de texte associé par l'attribut `"aria-labelledby"`

→ Le contenu de l'attribut `"aria-label"`

→ Le contenu du lien

→ Le contenu de l'attribut `title`

Le nom accessible doit absolument contenir l'intitulé visible afin de permettre aux logiciels de contrôle à la voix d'ouvrir les liens, **si ce n'est pas le cas l'activation du lien ne fonctionnera pas** (ex : intitulé visible : "toutes les actualités" / nom accessible : "cliquez sur toutes les actualités").

Pour les liens composés d'une seule image (image cliquable), il faudra **mettre un texte alternatif** (*alt=""*) qui sera l'intitulé du lien, il devra donc indiquer la destination du lien plutôt qu'une description de l'image.

Pour les liens composites (plusieurs contenus) il faudra **vérifier que la totalité du texte du lien soit explicite**, de plus il est conseillé d'élargir la zone de clic en CSS.

Contexte du lien

Il est facile d'utiliser un lien non explicite dépendant de son contexte (ex : "cliquez ici"), mais la navigation sera plus complexe.

Malgré tout **un lien non explicite devra avoir un contexte clair et précis** comme :

- Le contenu de la phrase ou du paragraphe (*<p>*) dans laquelle le lien est présent.
- Le contenu de l'item de liste (**) ou de l'item de liste parent dans lequel le lien est présent.
- Le contenu de la ou des cellules d'entête ou de la cellule dans laquelle le lien est présent.

Le lien devra malgré tout **être accessible** avec l'utilisation d'un attribut *"aria-describedby"*, avec l'attribut *"aria-label"* ou avec un *"title"* pour permettre une **lecture du contexte par les technologies d'assistance**.

Visibilité du lien

Un lien présent dans une phrase ou un paragraphe peut être difficile à distinguer, pour cela il faut **selon le RGAA** :

- Dans chaque page web, chaque lien dont la nature n'est pas évidente est-il visible par rapport au texte environnant ?

Il faut donc que le lien soit à la fois visible grâce à la couleur mais aussi grâce à une indication visuelle :

- La couleur du lien doit avoir un contraste supérieur ou = à 3:1 par rapport au texte environnant.
- Au survol une indication visuelle autre qu'un changement de couleur doit être présent (soulignage...)

SCRIPTS :

Les scripts sont des codes écrits sous forme de listes de commandes (par exemple en JavaScript), l'insertion se fait dans la balise <script>. Les sites web actuels sont composés de nombreux composants d'interfaces (menus, onglets, calendriers...),

Les composants d'interface doivent donc être accessibles à la fois pour les déficients visuels (compréhension du nom, fonction, changement d'état...) **mais aussi pour les handicapés moteurs** qui devraient pouvoir accéder à ces éléments sans l'usage de la souris. Ils doivent donc selon le RGAA :

→ Chaque script est-il, si nécessaire, compatible avec les technologies d'assistance ?

Pour concevoir une interface riche il faudra utiliser des gabarits de conception et/ou tester avec une base de référence.

Gabarits de conception ARIA (Design Pattern)

Les Gabarits de conception permettent aux développeurs de suivre un cahier des charges précis dans la conception de leurs interfaces. **Un gabarit de conception suit une structure précise :**

→ Description du composant d'interface

→ Un ou plusieurs exemples

→ Interactions clavier

→ Description des rôles / propriétés sur les éléments & changements d'états

Interfaces sans gabarit

En revanche, **il ne faut pas obligatoirement utiliser un gabarit de conception**, il est possible d'effectuer une conception aussi détaillée qu'avec un gabarit de conception et de la rendre accessible.

Il faudra tout de même **veiller à ce que les propriétés et les différents états soient correctement interprétés par les technologies d'assistance et par les navigateurs**.

Il est aussi possible pour les scripts de fournir une alternative accessible offrant le même niveau d'information (principalement sous forme textuelle).

Base de référence

Avec ou sans gabarit de conception, **il faut absolument tester les composants développés dans un environnement de test**, il faut ainsi **tester les configurations les plus utilisées** (sans tout tester). *Si la configuration utilisateur est connue il faut restreindre la base de référence à cette configuration.*

Ainsi pour qu'un élément soit considéré comme "**compatible à l'accessibilité**", il devra **être fonctionnel sur l'ensemble de la base de référence**.

Compatibilité au clavier (et aux dispositifs de pointage)

Les handicapés moteurs doivent pouvoir accéder à tous les éléments d'interface uniquement avec l'aide du clavier (sans l'utilisation de la souris), selon le RGAA :

→ Chaque script est-il contrôlable par le clavier et par tout dispositif de pointage ?

Ainsi il faut pouvoir accéder à ces éléments dans un ordre cohérent, permettant une navigation facile entre les différents éléments :

→ Avec **TAB** et **SHIFT + TAB** (éléments interactifs)

→ Avec la touche **ENTRÉE** ou **ESPACE** (boutons, case à cocher, liens...)

→ Avec la touche **ESPACE** puis avec les flèches pour sélectionner (liste déroulante)

Il faut tout de même veiller au focus puisque **certains scripts peuvent supprimer le focus** (suppression d'une ligne de tableau, soumission d'un formulaire...), ainsi **il faut que le focus soit positionné sur un élément logique** (ligne suivante, début de page...) avec l'utilisation de focus() en JavaScript.

Il faut aussi **veiller à ne pas avoir de piège au clavier**, c'est un composant qui bloque un utilisateur au clavier sans lui laisser la possibilité de le contourner, selon le RGAA :

→ Dans chaque page web, la navigation ne doit pas contenir de piège au clavier. Cette règle est-elle respectée ?

Ainsi la seule issue possible est le rechargement de la page. Malgré tout, cela est tout d'abord **très frustrant** mais **peut aussi bloquer certaines fonctionnalités du site web**. Il faudra ainsi veiller à ce que les pièges au clavier volontaires soient contournables (ex : fenêtre modale fermable avec le bouton "fermer" ou la touche échap).

Messages de statut & Zone Live

Les **Zones Live** sont des **composants mis à jour dynamiquement sans rechargement de la page** (en JavaScript), ils doivent être restitués aux technologies d'assistances via les attributs *"aria-live"* ou *"aria-atomic"*.

Les **messages de statut** sont des **zones lives permettant de notifier l'utilisateur**, ils sont ajoutés via l'attribut *"role"*. Ces messages doivent être brefs et ne doivent ni modifier le contexte ni déplacer le focus, ils sont soumis à une contrainte du RGAA :

→ Dans chaque page web, les messages de statut sont-ils correctement restitués par les technologies d'assistance ?

Il existe 4 types de messages de statuts :

→ ***role="status"*** : Informe sur la réussite ou le résultat d'une action sans alerte

(*aria-live="polite" aria-atomic="true"*),

→ ***role="alert"*** : Informe d'un message important, d'une suggestion ou d'une erreur

(*aria-live="assertive" aria-atomic="true"*),

→ ***role="log"*** : Indique l'ajout de nouvelles informations comme un journal de logs,

→ ***role="progressbar"*** : Informe l'utilisateur sur la progression d'une tâche

Changement de contexte

Un changement de contexte est un **changement majeur dans le contenu d'une page web**, ces changements concernent :

→ Les Changements d'**agent utilisateurs** (ouverture d'un fichier dans un autre logiciel par exemple)

→ Les Changements d'**espace de restitution** (affichage d'une nouvelle page / nouvel onglet)

→ Les Changements de focus

→ Les Changements de contenu dynamique dans une page (erreur d'un champ de formulaire par exemple)

Ces changements sont contraints à un critère du RGAA :

→ Pour chaque script qui initie un changement de contexte, l'utilisateur est-il averti ou en a-t-il le contrôle ?

ÉLÉMENTS OBLIGATOIRES :

Respect des standards

Une page web doit absolument **respecter certains standards** afin d'avoir un **code à la fois structuré mais aussi clair et facile à comprendre**. Le RGAA a donc imposé plusieurs critères :

- Chaque page web est-elle définie par un type de document ?
- Pour chaque page web, le code source généré est-il valide selon le type de document spécifié ?
- Dans chaque page web, les balises ne doivent pas être utilisées uniquement à des fins de présentation, cette règle est-elle respectée ?

Il faut ainsi **avoir un DOCTYPE valide au regard de la norme**, elle permet aussi la compatibilité entre les agents utilisateurs et les technologies d'assistance.
(Ex : `<!DOCTYPE html>`).

Il faudra ensuite selon le DOCTYPE déclaré valider le code source au regard de l'accessibilité, il faudra donc **valider le DOM**. Le code source généré devra cependant respecter :

- Respect des règles d'écriture par les balises / attributs / valeurs
- Imbrication conforme des balises (et ouverture / fermeture conforme)
- Unicité des valeurs d'attribut "id" dans la page
- Unicité des attributs sur un même élément.

Il faudra pour finir **respecter la sémantique** en veillant à ne pas utiliser des `<div>` ou `` avec plusieurs `
` pour créer des paragraphes, de plus il est conseillé d'utiliser une balise de titre pour donner de l'importance à un paragraphe.

Code de la langue et Sens de lecture:

Le code de langue d'une page web est défini par l'attribut "lang" ou "xml:lang" de la balise html. Le code de langue doit être conforme à la norme ISO 639-1, c'est-à-dire comporter **2 lettres pour le choix de la langue** (ex : "fr" (français), "en" (anglais), "de" (allemand) ...). Le RGAA a imposé plusieurs critères pour le code de langue :

- Dans chaque page web, la langue par défaut est-elle présente ?
- Pour chaque page web ayant une langue par défaut, le code de langue est-il pertinent ?
- Dans chaque page web, chaque changement de langue est-il indiqué dans le code source ?
- Dans chaque page web, le code de langue de chaque changement de langue est-il valide et pertinent ?

Il faudra ainsi veiller à ce que **lors d'un changement de langue, le code de langue soit indiqué dans le code source sur sa balise HTML**. De plus les logiciels de traduction braille pourront suivre le changement de langue et les synthèses vocales pourront les prononcer.

Il existe bien sûr des **cas particuliers** qui sont tolérés comme les **noms propres**, ou les **noms communs de langue étrangère utilisés dans le langage commun** (e-mail, week-end, hamburger...) ou les **termes issus d'une langue morte ou imaginaire** pour laquelle il n'existe pas d'interprétation vocale.

Certaines langues comme l'arabe ou l'hébreu se lisent de droite à gauche, or le sens de lecture par défaut sur une page web est de gauche à droite.

Ainsi lorsque l'on change de sens de lecture il faudra le spécifier dans le document :

- Sens de lecture de **gauche à droite** : *dir="ltr"* (**left to right**) (*PAR DEFAUT*)
- Sens de lecture de **droite à gauche** : *dir="rtl"* (**right to left**)

Titre de page

En plus de la langue, il y a un élément très important pour les sites web, c'est le titre de page. Le titre de page est le titre qui sera affiché dans le navigateur (dans l'onglet) et qui permettra lorsque l'on a plusieurs onglets de reconnaître rapidement le site web, il sera présent entre les balises `<title>...</title>` .Le RGAA impose des critères :

- Chaque page web a-t-elle un titre de page ? Si oui, est-il pertinent ?

Le titre de page est indispensable pour qu'un site web soit référencé puisque c'est la première information disponible, elle sert de base aux navigateurs pour effectuer l'historique, la mise en favori... Il ne faut donc pas mettre à jour autrement qu'après un rechargement de page.

STRUCTURATION DE L'INFORMATION :

Hiérarchiser les contenus

Il est indispensable de **structurer une page web** afin de pouvoir naviguer **rapidement sur la page** et de permettre lors de l'utilisation des outils d'assistance une **meilleure compréhension**. Ainsi le RGAA impose un critère :

→ Dans chaque page web, l'information est-elle structurée par l'utilisation appropriée de titres ?

Les différents titres (utilisation des balises <h...> (de 1 à 6)) permettent de **structurer les informations d'une page**, ils devront donc être relativement courts. De plus ils permettent aux aides techniques une navigation de titre en titre, cela permet une **navigation** à la fois **plus rapide** mais aussi répartie.

Ainsi il faut veiller à ce que les titres soient bien visibles et séparés des textes afin de **ne pas confondre texte grand en gras avec titre**. Il faut aussi veiller à bien **respecter les tailles de titre** (un titre de niveau 4 ne doit pas être plus grand qu'un titre de niveau 3), de plus **les titres ne doivent pas être anarchiques** (titre de niveau 3 avant un titre de niveau 2).

Afin d'avoir un rendu correct des titres pour les technologies d'assistance, il est conseillé les **attributs rôle="heading" et aria-level="1|2|...|6"**, ainsi si la balise est un div et n'est pas une balise <h...> elle sera quand même reconnue.

Structure du document

Pour pouvoir naviguer efficacement dans une page web, il est nécessaire d'identifier plusieurs zones de contenus :

- L'entête principal
- La navigation
- Le contenu principal
- Le pied de page

Le RGAA impose un critère pour la structure du document :

→ Dans chaque page web, la structure du document est-elle cohérente ?

- **L'entête principal** est la zone située en haut de la page web, ça peut être un **logo ou un slogan** par exemple.

L'entête est structuré avec une **balise <header> unique** dans la page. Pour faciliter l'accessibilité il sera conseillé de mettre un *role="banner"*.

- **Les zones de navigation** sont des zones que l'on retrouve sur toutes les pages du site web, elles permettront de **navigation entre les différentes pages du site**, il peut aussi y avoir des zones de navigation secondaires qui permettent de naviguer entre différentes sections.

Les navigations sont structurées avec une **balise <nav>**. Pour faciliter l'accessibilité il est conseillé d'utiliser un *role="navigation"* ainsi que d'un *aria-label="..."* afin d'indiquer à quoi sert la navigation.

- **La zone de contenu principal** est la zone contenant **tous les principaux contenus de la page** quels qu'ils soient.

Cette zone est structurée avec une **balise <main> unique** dans la page. Pour faciliter l'accessibilité il est conseillé d'utiliser un *role="main"* ainsi qu'un *role="complementary"* s'il y a besoin d'une zone de contenu supplémentaire.

- **Le pied de page** est la zone située en bas de la page web, il contient généralement les **informations générales relatives au site** (coordonnées, mentions légales...).

Cette zone est structurée avec une **balise <footer> unique** dans la page. Pour faciliter l'accessibilité il sera conseillé d'utiliser un *role="contentinfo"*.

Structure des listes

Il est nécessaire comme pour les titres de **structurer les contenus avec des éléments appropriés**, ainsi il faudra vérifier selon le RGAA :

→ Dans chaque page web, chaque liste est-elle correctement structurée ?

Ainsi pour les listes certain utilisateurs utilisant un clavier doivent pouvoir éviter une liste qui serait trop longue ou naviguer dans celle-ci en allant par exemple au début ou à la fin de cette dernière.

Il y a **deux types de liste** :

→ Les **listes ordonnées** : L'ordre de chaque élément est important pour la compréhension de la liste (utilisation des balises et)

→ Les **listes non-ordonnées** : L'ordre n'impacte pas la compréhension de la liste (utilisation des balises `` et ``)

Pour faciliter l'accessibilité, il est possible de *structurer les listes à l'aide des attributs* `role="list"` et `role="listitem"`. Mais il faudra veiller à ordonner la liste soi-même si besoin (à l'aide de numérotation par exemple).

Il existe aussi les **listes de définition** (`<dl>`), elles sont principalement utilisées pour décrire un terme ou réaliser une FAQ par exemple. On utilisera alors les **balises** `<dt>` et `<dd>` .

Citations

Les citations sont principalement **marquées par des guillemets**, il faut ainsi correctement indiquer la citation ce qui permettra aux outils d'assistance de bien la différencier des autres textes, ainsi le RGAA impose un critère pour les citations :

→ Dans chaque page web, chaque citation est-elle correctement indiquée ?

Il existe deux types de citations : les **citations en ligne** qui se trouvent **à l'intérieur d'un paragraphe** (balise `<q>`) et les **citations en bloc** qui forme un **élément indépendant** et donc peut être lu sans contexte (balise `<blockquote>`).

PRÉSENTATION DE L'INFORMATION :

Mise en forme et contenus

Mise en forme

Selon les différents handicaps, il est nécessaire de personnaliser l'affichage afin d'avoir une meilleure vision et compréhension des différents éléments de la page web, ainsi le RGAA impose :

→ Dans le site web, des feuilles de styles sont-elles utilisées pour contrôler la présentation de l'information ?

→ Dans chaque page web, le contenu visible porteur d'information reste-t-il présent lorsque les feuilles de styles sont désactivées ?

Il y a tout d'abord **certains éléments qui sont interdits** :

→ **Certaines balises** : basefont, blink, center, font, marquee, s, strike, tt et big

→ **Certains attributs** : align, alink, background, bgcolor, border, cellpadding, cellspacing, char, charoff, clear, compact, color, frameborder, hspace, link, marginheight, marginwidth, text, valign, vlink, vspace, size.

Les attributs width et height sont aussi interdits pour les autres balises que ``, `<object>`, `<canvas>` et `<svg>`.

Contenus

Le fond de la page est important pour un contraste efficace avec le texte, mais lorsque l'on insère une image de fond en CSS, elle ne doit pas être porteuse d'information, de plus selon le RGAA :

→ Dans chaque page web, le contenu visible porteur d'information reste-t-il présent lorsque les feuilles de style sont désactivées ?

→ Dans chaque page web, l'information reste-t-elle compréhensible lorsque les feuilles de styles sont désactivées ?

Ainsi les utilisateurs ayant besoin de personnaliser l'affichage d'une page devront pouvoir voir tous les contenus porteurs d'information peu importe le type d'affichage.

De plus **l'ordre des éléments affichés doit absolument correspondre à l'ordre du DOM**.

Agrandissement du texte

Contexte

Certains déficients visuels ou dyslexiques ont **besoin d'agrandir le texte ou l'interface et/ou besoin de personnaliser le texte** comme la police d'écriture, la couleur de fond (et du texte), l'espacement entre les caractères... Pour cela le RGAA a imposé plusieurs critères :

- Dans chaque page web, le texte reste-t-il lisible lorsque la taille des caractères est augmentée jusqu'à 200% au moins ?
- Pour chaque page web, les contenus peuvent-ils être présentés sans avoir recours à un défilement vertical pour une fenêtre ayant une hauteur de 256px ou à un défilement horizontal pour une fenêtre ayant une largeur de 320px ?
- Dans chaque page web, les propriétés d'espacement du texte peuvent-elles être redéfinies par l'utilisateur sans perte de contenu ou de fonctionnalité ?

Contraintes

Il faut ainsi que **la mise en forme reste cohérente, lisible et compréhensible** malgré les modifications dont :

- Zoom du texte ou zoom graphique à 200%
- Utilisation d'un composant d'interface qui permet d'agrandir le texte ou zoom

Il faudra aussi **éviter les barres de défilement** sauf pour les images, graphiques, vidéos, présentations, jeux ou tableaux.

Lors de l'utilisation de la balise meta viewport, il faut veiller à ne pas utiliser d'*user-scalable="no"* ou de *maximum-scale* inférieur à 2.0 afin de ne pas empêcher le zoom.

Conseils

Voici quelques **conseils de mise en place de l'agrandissement du texte** :

- Ne pas limiter la hauteur des éléments avec "height" (préférer "*min-height*")
- Utiliser des unités relatives pour les dimensions ("*em*", "*rem*", "%" ...)
- Limiter l'utilisation de "*fixed*"
- Utiliser les media queries pour définir les points de rupture

Couleurs et fonds

Tout élément doit avoir une couleur de texte ainsi qu'une couleur de fond, il faut ainsi respecter la contrainte du RGAA :

→ Dans chaque page web, les déclarations CSS de couleurs de fond d'élément et de police sont-elles correctement utilisées ?

Il est possible de vérifier cela en modifiant les couleurs par défaut dans le navigateur par exemple.

Focus et contenus cachés

La visibilité du focus est **cruciale pour les utilisateurs ne pouvant pas utiliser la souris**, ainsi le RGAA impose :

→ Dans chaque page web, pour chaque élément recevant le focus, la prise de focus est-elle visible ?

Le focus peut être personnalisé (*focus & outline* en CSS) mais il faut veiller à ce qu'il ne soit pas dégradé (*interdiction : outline: 0 | none | transparent*). De plus il faudra que **le contraste entre le focus et l'arrière-plan soit supérieur ou = à 3:1**.

Certains contenus peuvent être cachés (*display none, visibility hidden...*), cependant il faut pouvoir vérifier si ces contenus doivent être aussi **ignorés par les outils d'assistance** ou non.

Par exemple une image survolée / focusée affichant du texte (avec *display none*) pourra être ignorée (le texte sera restitué aux aides techniques cependant). Pour un menu par exemple masqué qui s'affiche au clic sur un bouton il devra être ignoré par les technologies d'assistance.

Informations et contrôles

L'information doit être donnée globalement, comme vu précédemment pour la couleur elle **ne doit pas être donnée uniquement par un élément**, l'information doit donc selon le RGAA :

→ Dans chaque page web, l'information ne doit pas être donnée uniquement par la forme, taille ou position. Cette règle est-elle respectée ? Est-elle implémentée de façon pertinente ?

→ Dans chaque page web, les contenus additionnels apparaissant à la prise de focus ou au survol d'un composant d'interface sont-ils contrôlables par l'utilisateur ?

→ Dans chaque page web, les contenus additionnels apparaissant via les styles CSS uniquement peuvent-ils être rendus visibles au clavier et par tout dispositif de pointage ?

Lors du zoom le contenu additionnel doit absolument rester lisible et doit être contrôlable pour pouvoir lire ce qu'il cache, ainsi il sera possible de le consulter même lors de la navigation au clavier.

FORMULAIRES :

Étiquette de champ

L'**étiquette de champ** est **obligatoire** pour les formulaires, elle doit absolument être pertinente pour connaître la nature, le type ou le format des informations attendues. De plus elle permet aux déficients visuels de naviguer entre les champs.

Le RGAA impose des critères :

- Chaque champ de formulaire a-t-il une étiquette ? Si oui, est-elle pertinente ?
- Dans chaque formulaire, chaque étiquette associée à un champ de formulaire ayant la même fonction et répétée plusieurs fois dans une même page ou dans un ensemble de pages est-elle cohérente ?
- Dans chaque formulaire, chaque étiquette de champ et son champ associé sont-ils accolés ?

Il y a **plusieurs moyens de lier une étiquette à un champ** :

- Ajouter l'attribut *"for"* (avec un id) dans la balise `<label>`
- Ajouter un attribut *"aria-labeledby"* (avec un id)
- Ajouter un attribut *"aria-label"*
- Ajouter un attribut *"title"*.

Il est conseillé d'utiliser une **étiquette visible associée au champ**, si l'étiquette est tout de même invisible, un texte visible (*label*, *aria-label*, *title*) doit pouvoir permettre de comprendre la nature de la saisie qui est attendue.

Sur une **fonctionnalité identique** ou des champs demandant la même information, il faut veiller à ce que les **étiquettes** soient **différentes** mais renseigner à l'utilisateur qu'il faut les remplir de la même manière afin d'éviter toute confusion.

Il faut veiller à **accoler obligatoirement l'étiquette au champ correspondant** (Le sens peut être inversé selon le sens de la langue) :

- Pour les champs il faut qu'elle soit positionnée à gauche ou au-dessus du champ.
- Pour les cases à cocher ou les boutons radio à droite ou en dessous.

Regroupement de champs

Afin d'éviter les incompréhensions en cas d'étiquettes identiques mais de nature différente, il est important de pouvoir **regrouper les champs** (zones à remplir, boutons radio, cases à cocher...), ce qui permettra une compréhension plus rapide des informations à saisir. On utilisera "*field-wrap*". On vérifie alors plusieurs critères du RGAA :

- Dans chaque formulaire, les champs de même nature sont-ils regroupés ?
- Dans chaque formulaire, chaque regroupement de champs de même nature a-t-il une légende ?
- Dans chaque formulaire, chaque légende associée à un regroupement de champs de même nature est-elle pertinente ?
- Dans chaque formulaire, les items de même nature d'une liste sont-ils regroupés de manière pertinente ?

Les listes aussi peuvent être regroupées et/ou rangées, afin de *créer une hiérarchie* et un *ordre logique*, pour cela l'**élément "select" propose une balise "optgroup"** pour grouper les différentes options, on peut même grâce à l'attribut label apporter une étiquette au groupe. Il est aussi possible de ne pas utiliser l'élément select, une balise ayant l'attribut *role="listbox"* pourra utiliser l'attribut *role="option"* (mais le regroupement ne sera pas possible).

Boutons

Afin d'avoir une compréhension rapide par les différents utilisateurs, **il est obligatoire d'utiliser un intitulé visible, pertinent et clair sur l'action du bouton** (ne pas utiliser "Ok", "Envoyer"...). Il faut ainsi respecter le critère du RGAA :

- Dans chaque formulaire, l'intitulé de chaque bouton est-il pertinent ?

Ainsi lors de la navigation entre les boutons par les technologies d'assistance, **les intitulés permettront de comprendre totalement l'action réalisée par le bouton**. Les formulaires réalisant les mêmes actions (recherche par exemple) mais avec des objectifs différents devront avoir des intitulés différents afin de ne pas les confondre.

Il y a trois types de boutons :

- La balise `<input>` avec l'attribut *type= "submit" / "reset" / "button"*
- La balise `<input>` avec l'attribut *type="image"*
- La balise `<button>` avec l'attribut *type= "submit" / "reset" / "button"*

Ainsi il est possible **d'utiliser plusieurs éléments pour l'intitulé du bouton**, qui seront obtenus selon l'ordre suivant :

- Le passage de texte associé à l'attribut *aria-labelledby*
- Le contenu de l'attribut *aria-label* (notamment pour les outils d'assistance)
- L'alternative *alt="..."* (pour les balises *image*)
- L'attribut *value="..."* (pour la balise `<input>`)
- Le contenu de la balise `<button>`.
- Le contenu de l'attribut *title*

Contrôle de saisie

Lors de la vérification des saisies dans les formulaires, il est indispensable **d'indiquer de façon explicite la saisie obligatoire**, selon le RGAA :

- Dans chaque formulaire, le contrôle de saisie est-il utilisé de manière pertinente ?
- Dans chaque formulaire, le contrôle de saisie est-il accompagné, si nécessaire, de suggestions facilitant la correction des erreurs de saisie ?

Ainsi il faudra absolument **prévenir l'utilisateur lorsque la saisie sera obligatoire**, il y a **plusieurs moyens possibles** :

- Utilisation d'une étoile "*" en prévenant l'utilisateur de son sens (* = champ obligatoire)
- Un texte positionné avant le(s) champ(s) (ex : les champs ... sont obligatoires)
- Indication directement dans le champ concerné (ex : Nom (obligatoire) :)

Pour ces cas-là il est possible d'utiliser aussi les attributs "required" ou "aria-required" sur le champ concerné. Mais pour éviter la redondance il est conseillé de le masquer.

Pour les champs de saisie dans un format spécifique, il est possible d'apporter à l'utilisateur une **aide à la saisie**. Cela marche pour les emails ou le code postal par exemple, mais pour les champs comportant un masque de saisie il ne faudra pas avoir d'aide présente (type date / time / datetime).

Lors des erreurs de saisie l'utilisateur devra être prévenu de la même façon (même pour des champs non obligatoires),

Il est possible d'implémenter l'aide de saisie ou l'erreur de saisie **directement dans la balise <label> ou dans un passage de texte visible associé avec "aria-describedby"** (rempli avec

l'id du passage de texte). De plus pour les **erreurs de saisie** il est possible d'**utiliser l'attribut *aria-invalid="true"***.

Données

Afin d'**éviter que des informations sensibles soient envoyées sans aucune vérification**, il faut respecter le critère suivant :

→ Pour chaque formulaire qui modifie ou supprime des données, ou qui transmet des réponses à un test ou à un examen, ou dont la validation a des conséquences financières ou juridiques, les données saisies peuvent-elles être modifiées, mises à jour ou récupérées par l'utilisateur ?

Ainsi **l'utilisateur devra pouvoir** :

→ **Vérifier et modifier les informations avant un envoi final** lors de la soumission d'un formulaire

→ Lorsque le formulaire a plusieurs étapes, **revenir à toutes les étapes du processus**

→ Lorsqu'il n'est pas possible de modifier les données, accéder à une étape supplémentaire de validation (case à cocher par exemple).

Afin de permettre un remplissage plus rapide des champs de saisie, il est possible d'utiliser un **remplissage automatique** des différents champs à l'aide de l'attribut "*autocomplete*" (si l'utilisateur l'autorise et l'enregistre dans son navigateur), malgré tout il faudra vérifier ce critère du RGAA :

→ La finalité d'un champ de saisie peut-elle être déduite pour faciliter le remplissage automatique des champs avec les données de l'utilisateur ?

NAVIGATION :

Moyens de navigation

En fonction des différents besoins des utilisateurs, il existe différents **moyens de navigation**, chaque ensemble de pages devra donc utiliser **deux moyens de navigation** parmi la **navigation principale**, le **plan du site** et la **recherche**.

Il faudra cependant qu'ils respectent ces critères du RGAA :

→ Chaque ensemble de pages dispose-t-il de deux systèmes de navigation différents, au moins ?

→ Dans chaque ensemble de pages, le menu et les barres de navigation sont-ils toujours à la même place ?

→ Dans chaque ensemble de pages, la page "plan du site" est-elle atteignable de manière identique ?

→ Dans chaque ensemble de pages, le moteur de recherche est-il atteignable de manière identique ?

Les différents moyens de navigations devront **obligatoirement être situés au même endroit sur chaque page** de l'ensemble afin de ne pas perdre les utilisateurs. Il peut n'y avoir qu'un seul moyen de navigation pour les petits sites.

Le plan du site doit représenter l'architecture globale du site, il doit donc permettre d'atteindre l'ensemble des pages. **La recherche, elle, doit permettre de rechercher l'ensemble des contenus** présents sur le site.

Lien d'évitement

Chaque page web doit posséder un lien d'évitement afin d'éviter d'accéder à la zone de contenu, ce lien doit être situé **à la même place sur toutes les pages** et être visible ou visible à la prise de focus. Ce qui validera la contrainte du RGAA :

→ Dans chaque page web, un lien d'évitement ou d'accès rapide à la zone de contenu principal est-il présent ?

Regroupement et tabulation

Comme vu précédemment, ARIA permet de définir le rôle "**Landmark**" à des **zones principales du document**, ces zones ne doivent pas être cachées, il faudra donc penser à entrer les différents rôles pour les différents "*Landmark*" du document, il faudra, selon le RGAA :

→ Les zones de regroupement de contenus présentes dans plusieurs page web (zones d'en tête, de navigation principale, de contenu principal, de pied de page et de moteur de recherche) peuvent-elles être atteintes ou évitées ?

Afin de naviguer de façon compréhensible, il est indispensable de **s'assurer que l'ordre de tabulation reste logique et cohérent**, ainsi selon le RGAA :

→ Dans chaque page web, l'ordre de tabulation est-il cohérent ?

Contenus additionnels

Les contenus additionnels comme les descriptions qui sont présentées à l'utilisateur via un **composant d'interface** à la prise de focus ou au survol doivent être **accessibles au clavier**, selon le RGAA :

→ Dans chaque page web, les contenus additionnels apparaissant au survol, à la prise de focus ou à l'activation d'un composant d'interface sont-ils si nécessaire atteignables au clavier ?

CONSULTATION :

Limite de temps

Certaines actions réalisées sur la page web ont une **limite de temps** avant de s'effectuer comme le rafraichissement des informations d'un site ou **un temps imparti** pour remplir certaines informations. Le RGAA impose donc un critère pour ces différentes actions :

→ Pour chaque page web, l'utilisateur a-t-il le contrôle de chaque limite de temps modifiant le contenu ?

Il faudra ainsi que **l'une de ces règles soit implémentée** pour permettre d'**aider les utilisateurs ayant un handicap** et ayant donc besoin de passer plus de temps sur la page (non applicable pour les examens par exemple) :

→ Pouvoir **arrêter ou relancer le rafraichissement**

→ Pouvoir **augmenter la limite de temps entre deux rafraichissements** de 10x (minimum)

→ **Avertir de l'imminence du rafraichissement** 20 secondes avant et permettre d'augmenter sa limite

→ **Limite de temps** entre deux rafraichissements de 20 heures

Nouvelles fenêtres

L'ouverture de nouvelles fenêtres est interdite sans action de l'utilisateur. L'ouverture d'une fenêtre sans avertir au préalable pourrait perturber ou perdre certains utilisateurs, ainsi selon le RGAA :

→ Dans chaque page web, l'ouverture d'une nouvelle fenêtre ne doit pas être déclenchée sans action de l'utilisateur. Cette règle est-elle respectée ?

Téléchargements

Les **documents téléchargés** sur la page web doivent être tout aussi **accessibles** que cette dernière, ainsi il faudra respecter ces critères :

→ Dans chaque page web, chaque document bureautique en téléchargement possède-t-il, si nécessaire, une version accessible ?

→ Pour chaque document bureautique ayant une version accessible, cette version offre-t-elle la même information ?

Le **même niveau d'information** devra être accessible aux utilisateurs, si ce n'est pas possible un lien vers la/les pages de contenu web accessible ou un fichier dans un autre format devra être fourni.

Contenu Cryptique

Tout comme les différents éléments présents sur la page web, **le contenu cryptique devra posséder une alternative** afin de permettre aux outils d'assistance de lire ces éléments, le RGAA impose donc des critères pour le contenu cryptique :

→ Dans chaque page web, chaque contenu cryptique (art ASCII, émoticône, syntaxe cryptique) a-t-il une alternative ? Si oui est-elle pertinente ?

Gestes complexes

Certains utilisateurs ayant un handicap pourraient **ne pas réussir à effectuer des gestes complexes** (drag and drop par exemple), il faudra ainsi **permettre aux utilisateurs de réaliser ces actions avec des gestes plus simples** (clic de la souris...) ainsi cela permettra de répondre au critère du RGAA :

→ Dans chaque page web, les fonctionnalités utilisables ou disponibles au moyen d'un geste complexe peuvent-elles être également disponibles au moyen d'un geste simple ?

Annuler une action

Le dispositif de pointage peut, lorsqu'il est utilisé, engendrer plusieurs **erreurs par inadvertance** (précision), ainsi il faudra selon le RGAA :

→ Dans chaque page web, les actions déclenchées au moyen d'un dispositif de pointage sur un point unique de l'écran peuvent-elles faire l'objet d'une annulation ?

Lorsqu'un utilisateur déclenchera une action sur un élément interactif par inadvertance, il devra **pouvoir l'annuler directement** en éloignant le pointeur de la cible par exemple. Cela marchera uniquement sur les évènements relâché ou relevé. Pour ce qui est du glisser/déposer il devra être annulable en déposant à un endroit non prévu par exemple.

Orientation de l'écran

Certains utilisateurs du site web pourraient avoir **besoin d'utiliser le site avec une orientation portrait ou paysage selon les besoins**. Il faudra ainsi que le contenu de la page soit responsive pour permettre une utilisation du site web peu importe l'orientation de la page. Cela permet de répondre au critère du RGAA :

→ Dans chaque page web, le contenu proposé est-il consultable quelle que soit l'orientation de l'écran (portrait ou paysage) ?

Changements et mouvements

Une page web peut comporter des **changements brusques visuels**, malgré tout ils devront respecter certains critères du RGAA :

→ Dans chaque page web, les changements brusques de luminosité ou les effets de flash sont-ils correctement utilisés ?

→ Dans chaque page web, chaque contenu en mouvement ou clignotant est-il contrôlable par l'utilisateur ?

Les **changements brusques de luminosité ou d'effets de flash** sont **fortement déconseillés** puisqu'ils peuvent provoquer une gêne voir des crises d'épilepsie pour les utilisateurs.

Si malgré tout ces changements brusques sont contenus sur la page, ils devront respecter ces règles :

→ Fréquence de l'effet < 3 par seconde

→ Surface totale cumulée inférieure ou = à 21824 pixels

Les **contenus en mouvement ou clignotant** peuvent aussi perturber les utilisateurs notamment les hyperactifs, ce qui pourrait les empêcher de consulter le reste de la page.

Ces changements devront notamment respecter ces règles s'ils sont présents sur les pages web :

→ Durée du mouvement <= à 5 secondes (sans répétition)

→ Arrêt ou reprise du mouvement par l'utilisateur

→ (Dés)Affichage du contenu en mouvement

→ Affichage de la totalité des informations sans mouvement

Certaines interfaces notamment mobiles permettent de réaliser des actions qui **suivent le mouvement de l'appareil**, mais certaines utilisateurs pourraient ne pas pouvoir effectuer ces actions, il faudra ainsi selon le RGAA :

→ Dans chaque page web, les fonctionnalités qui impliquent un mouvement de l'appareil peuvent-elles être satisfaites de manière alternative ?

Il faudra ainsi **prévoir des contrôles alternatifs simples** et ainsi permettre un **bouton pour (dés)activer le mouvement de l'appareil**.