

Rapport d'audit accessibilité pour le site CCAS de Lille

Rapport d'audit RGAA pour **le site CCAS de Lille**

11/02/2025 – Version 1.0
Propriété EOLAS – Confidentiel



Suivi des modifications

Version	Date	Auteur	Motif
1.0	11/02/2025	Laura Gauducheau	Elaboration du rapport d'audit



Table des matières

1. Introduction	5
1.1 Accessibilité du site	6
1.2 Contenus dérogés	7
1.3 Note sur le relevé des non-conformités.....	7
1.4 Obligation d’affichage	7
1.4.1 Mention sur la page d’accueil.....	7
1.4.2 Déclaration d’accessibilité.....	8
1.4.3 Schéma pluriannuel de mise en accessibilité.....	8
2. Images	8
2.1 Images porteuses d’information	8
2.2 Images de décorations	9
2.3 Images CAPTCHA.....	10
3. Couleurs	11
3.1 Contrastes des textes	11
3.2 Contrastes des composants d’interfaces.....	12
4. Tableaux.....	13
4.1 Tableaux de mise en forme.....	14
5. Liens.....	14
6. Scripts	15
6.1 Disclosure (Afficher / Cacher).....	17
6.2 Tab.....	18
6.3 Changements de contexte	18
7. Eléments obligatoires.....	18
7.1 Indication du type de document et validité du code HTML	19
7.2 Balises utilisées uniquement à des fins de présentation	19
8. Structuration de l’information.....	19
9. Présentation de l’information	20



9.1	Utilisation exclusive de CSS pour la présentation.....	20
9.2	Visibilité du focus.....	20
9.3	Information donnée par la forme, la taille ou la position	21
10.	Formulaires	22
10.1	Contrôles de saisie et aide à la saisie	22
10.2	Indications de la nature des saisies (autocomplete)	23
11.	Navigation.....	25
11.1	Landmarks	25
11.2	Ordre de tabulation	26
12.	Consultation	26

1. Introduction

La version utilisée pour réaliser les tests est la version 4.1.2 du RGAA.

Les technologies utilisées sur le site sont les suivantes :

- HTML5
- CSS
- Javascript

Les outils suivants ont été utilisés pour vérifier l'accessibilité :

- WCAG Contrast Color Checker
- Stylus
- Web Developer Toolbar
- headingsMap
- Inspecteur du navigateur

La restitution des contenus avec les technologies d'assistance a été testée conformément à l'environnement de test décrit dans le RGAA 4.1.2 :

- NVDA 2024+ Edge
- NVDA 2024+ Firefox

Echantillon

L'audit a porté sur un échantillon de 13 pages.

Echantillon		
Page	Titre	Url
P01	Accueil	https://solidarites.lille.fr/
P02	Authentification	https://solidarites.lille.fr/8-inscription.htm
P03	Contact	https://solidarites.lille.fr/5-contact.htm
P04	Accessibilité	https://solidarites.lille.fr/21-accessibilite.htm
P05	Mentions légales	https://solidarites.lille.fr/15-mentions-legales.htm
P06	Echanger avec le webmestre	https://solidarites.lille.fr/acteur/831/3-eolas.htm
P07	Plan du site	https://solidarites.lille.fr/17-plan-de-site.htm



P08	Modifier ma fiche	https://solidarites.lille.fr/discussion/3-annuaire-des-organismes-de-l-action-sociale-a-lille.htm?ACM_SUJET=DESC
P09	Liste de l'annuaire	https://solidarites.lille.fr/discussion/3-annuaire-des-organismes-de-l-action-sociale-a-lille.htm?ACM_SUJET=DESC
P10	Fiche d'un annuaire	https://solidarites.lille.fr/acteur/868/3-0-800-59-11-11-numero-academie-du-nord-harcelement-scolaire.htm
P12	Fiche activité	https://solidarites.lille.fr/aide/903/2-victimes-de-violences-a-l-ecole.htm?show=2180
P13	Modification de mes coordonnées	https://solidarites.lille.fr/modification/3-annuaire-des-organismes-de-l-action-sociale-a-lille.htm#tpl_coord

Une grille d'audit est annexée à ce rapport, elle contient les résultats du contrôle de chaque page de l'échantillon au regard des critères de contrôle RGAA applicables.

1.1 Accessibilité du site

Le niveau global de conformité relevé atteint **57,6%** de conformité sur l'ensemble des pages auditées.

Le taux moyen de conformité s'élève à 84%.

Le site est globalement relativement conforme aux critères d'accessibilité, mais présente plusieurs points faibles majeurs qui impactent fortement l'expérience des utilisateurs en situation de handicap.

Les non-conformités les plus bloquantes sont :

- Accessibilité des images : De nombreuses images manquent d'alternatives textuelles pertinentes ou en sont dotées alors qu'elles ne comportent pas d'information, rendant l'information visuelle inaccessible aux utilisateurs aveugles ou malvoyants.
- Accessibilité des scripts : Certains scripts interactifs ne sont pas correctement pris en charge par les technologies d'assistance, rendant certaines fonctionnalités difficiles, voire impossibles, à utiliser pour les personnes naviguant au clavier ou avec un lecteur d'écran.
- Problèmes de navigation : L'organisation des contenus et des interactions présente des incohérences (absence de liens d'évitement, repères ARIA mal implémentés), ce qui nuit à une navigation fluide et compréhensible.

- Consultation du site : Certains contrastes de texte sont insuffisants, et la mise en page peut entraîner des difficultés de lecture, notamment pour les utilisateurs ayant des troubles cognitifs ou visuels.

Les principales personnes impactées par ces non-conformités sont les utilisateurs aveugles, malvoyants, ayant des troubles moteurs ou cognitifs, ainsi que ceux navigant exclusivement au clavier.

1.2 Contenus dérogés

La dérogation est un moyen d'exclure du champ d'application du référentiel des contenus particuliers. Les cas de dérogations sont très encadrés, vous en trouverez le détail dans le [guide des dérogations](#).

Les dérogations émises notamment pour charge disproportionnée demandent en contrepartie la mise en place d'un moyen de compensation pour les utilisateurs.

1.3 Note sur le relevé des non-conformités

Ne sont cités dans ce rapport que quelques exemples issus du relevé des non-conformités (tableau joint).

De plus, le relevé d'audit mentionne les différents types de non-conformités, mais ne cite pas toutes les occurrences. Toutes les erreurs relevées doivent donc être accompagnées, en phase de correction, d'une vérification attentive afin de détecter toutes les occurrences d'une non-conformité sur chaque page du site.

1.4 Obligation d'affichage

Nous rappelons ici les obligations d'affichage définies par le [décret n° 2019-768 du 24 juillet 2019](#) relatif à l'accessibilité aux personnes handicapées des services de communication au public en ligne.

Le défaut de publication est passible d'une sanction financière de 20 000 € par publications, par site et par an.

1.4.1 Mention sur la page d'accueil

Sur la page d'accueil au moins, vous êtes tenu d'afficher ce niveau. Cette mention peut être cliquable et conduire vers la page Accessibilité ou vers la déclaration d'accessibilité. Elle doit être rédigée sur le modèle : « **Accessibilité : partiellement conforme** »

1.4.2 Déclaration d'accessibilité

Vous êtes tenu de publier [une déclaration d'accessibilité](#). Le RGAA 4 vous en fournit un modèle et le détail.

1.4.3 Schéma pluriannuel de mise en accessibilité

Vous devez également mettre en place un schéma pluriannuel d'accessibilité et des plans annuels de mise en application de ce schéma. [Le RGAA 4 en définit les contours et les éléments à détailler.](#)

Ces documents doivent être accessibles depuis les déclarations d'accessibilité des sites. À titre d'exemple, consultez le [Schéma pluriannuel de mise en accessibilité de la région Île-de-France et son plan annuel.](#)

2. Images

Donner à chaque [image porteuse d'information](#) une [alternative textuelle](#) pertinente et une [description détaillée](#) si nécessaire. Lier les [légendes](#) à leurs images. Remplacer les [images textes](#) par du [texte stylé](#) lorsque c'est possible.

2.1 Images porteuses d'information

Pour créer une alternative aux images, il existe plusieurs méthodes, listées ci-dessous.

Pour les balises ``, `<area>`, `<input type="img">` :

- attribut `alt="[alternative]"` ;
- propriété `aria-labelledby="ID-texte-alternatif"` ;
- propriété `aria-label="[alternative]"` ;
- attribut `title="[alternative]"`.

Pour les balises `<svg>` :

- attribut `role="img"` et propriété `aria-label="[alternative]"` ;
- propriété `aria-labelledby="ID-texte-alternatif"` ;
- attribut `title="[alternative]"`.

Pour les balises `<object>` de `type="image"` ou `<embed>` de `type="image"` ou `<canvas>` :

- propriété `aria-label="[alternative]"` ;
- propriété `aria-labelledby="ID-texte-alternatif"` ;
- attribut `title="[alternative]"`.

Note : la propriété `aria-labelledby` est utilisée pour créer une liaison entre le passage de texte qui fait office d'alternative et l'image sur le modèle :

```
<p id="id-texte">>[texte alternatif]</p>

```

Constat



Correction

- Pour remplir correctement l'alternative de l'image, il faut reprendre au minimum les informations textuelles visibles.
- Vérifier qu'il n'existe pas, dans les pages ou les écrans, d'autres images porteuses d'informations sans alternatives.

2.2 Images de décorations

Une image de décoration ne contient aucune information indispensable à la compréhension du contenu auquel elle est associée. Il est important que ces images ne soient pas restituées par les technologies d'assistance (ex. lecteur d'écran).

Note : la propriété `aria-hidden="true"` empêche la restitution par les technologies d'assistance, elle doit être utilisée sur les balises `<svg>`, `<object>`, `<embed>`, `<canvas>` pour marquer les images comme décoratives.

Constat

	<p>Cette image possède un alt rempli alors qu'elle ne donne aucune information essentielle à l'utilisateur.</p>
	

Correction

- Utiliser un attribut `alt=""`
- Vérifier qu'il n'existe pas dans la page d'autres images similaires.

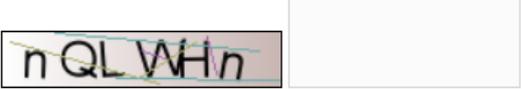
2.3 Images CAPTCHA

Un CAPTCHA est un test utilisé pour distinguer un utilisateur humain d'un ordinateur. Le test utilise souvent des images contenant du texte déformé, mélangé avec d'autres formes ou utilisant des jeux de couleur altérée, que l'utilisateur est invité à retaper. D'autres formes de CAPTCHA peuvent être basées sur des questions logiques ou des extraits sonores.

Pour chaque image CAPTCHA, une alternative doit être présente pour identifier la nature et la fonction de l'image. Et une solution d'accès alternative au contenu est présente. Il faut donc au moins :

- Soit un accès à une autre forme de CAPTCHA non graphique
- Soit une autre solution permettant d'accéder à la fonctionnalité

Constat

<p>Sécurité</p> <p>Captcha *</p> <div data-bbox="204 347 726 436">  </div> <p>(Regénérer)</p>	<p>Aucune solution alternative n'est proposée.</p>
--	--

Correction

- Proposer une autre forme de CAPTCHA non graphique
- Il existe une autre solution d'accès à la fonctionnalité qui est sécurisée par le CAPTCHA
- Vérifier qu'il n'existe pas, dans les pages ou les écrans, d'autres images CAPTCHA qui nécessiteraient une alternative.

3. Couleurs

Ne pas donner l'[information](#) uniquement par la couleur et utiliser des [contrastes](#) de couleurs suffisamment élevés pour les textes, les [composants d'interface](#) ou les éléments porteurs d'informations.

3.1 Contrastes des textes

Certaines personnes perçoivent peu ou mal les couleurs ou les associations de couleurs, il est donc nécessaire de fournir des textes avec un rapport de contraste suffisant par rapport au fond.

- Pour les textes qui ont une taille de police calculée **inférieure à 24px sans effet de graisse** ou une taille de police calculée **inférieure à 18,5px avec effet de graisse**, le rapport de contraste entre la couleur du texte (y compris le texte en image) et son arrière-plan doit être de **4.5:1**, au moins.
- Pour les textes qui ont une taille de police calculée **supérieure ou égale à 24px sans effet de graisse** ou une taille de police calculée **supérieure ou égale de 18,5px avec effet de graisse**, le rapport de contraste entre la couleur du texte (y compris le texte en image) et son arrière-plan doit être de **3:1**, au moins.

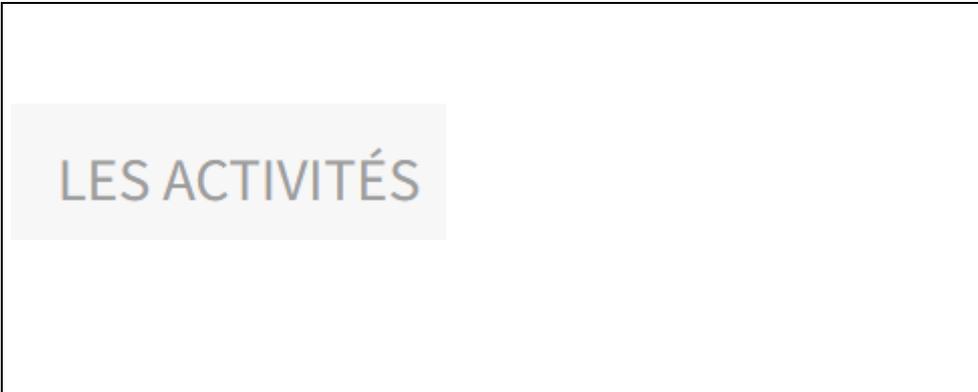


Vous pouvez contrôler les contrastes grâce à l’extension Firefox ou Chrome [WCAG Contrast Checker](#) ou l’outil [Colour Contrast Analyser](#).

S’il est impossible de modifier les couleurs, vous pouvez mettre en place un mécanisme alternatif pour obtenir des contrastes suffisants. Ce mécanisme alternatif doit être accessible et suffisamment contrasté.

Vous pouvez voir un exemple de ce type de mécanisme sur le site [AccessConfig](#). Dans les réglages d’accessibilité, vous trouverez une option qui permet d’afficher le site avec des contrastes renforcés et inversés.

Constat

	<p>Sur la page « Echanger avec un webmestre », le titre "Les activités" #9D9D9D sur fond F7F7F7 avec un ratio de 2,53 possède un contraste insuffisant.</p>
	<p>Sur la page « Fiche annuaire », au hover sur le texte bleu #034974 sur fond #735287 au hover, a un ratio de 1,5</p>

Correction

- Modifier la couleur de police ou de fond d’élément pour obtenir un rapport de contraste de 4.5, au moins.
- Autrement, proposer un mécanisme alternatif pour afficher une version suffisamment contrastée de la page.
- Vérifier qu’il n’existe pas, dans les pages ou les écrans, d’autres textes ayant un rapport de contraste insuffisant.

3.2 Contrastes des composants d’interfaces



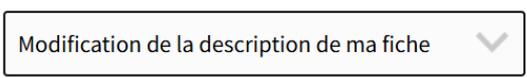
Un composant d'interface est un élément avec lequel l'utilisateur peut interagir, par exemple un bouton, un lien, une zone de saisie. Certains composants peuvent être plus complexes comme un menu, une fenêtre de dialogue, un système d'onglets. Enfin un composant d'interface peut être basé sur des éléments natifs de HTML ou développés de toutes pièces en JavaScript et des attributs WAI-ARIA. En particulier pour les éléments ayant des attributs WAI-ARIA correspondant à un [motif de conception](#) il est recommandé de considérer le document [WAI-ARIA 1.1 Authoring Practices](#) lors de leur implémentation.

Certaines personnes perçoivent peu ou mal les couleurs ou les associations de couleurs, il est donc nécessaire de fournir des composants d'interface avec un rapport de contraste suffisant de **3:1** par rapport à leur environnement et leurs différents états.

Vous pouvez contrôler les contrastes grâce à l'extension Firefox ou Chrome [WCAG Contrast Checker](#) ou l'outil [Colour Contrast Analyser](#).

S'il n'est pas possible de respecter les contrastes de couleurs, proposer un mécanisme permettant de modifier les couleurs pour obtenir des contrastes suffisants.

Constat

	<p>La flèche de la liste déroulante n'est pas suffisamment contrastée #C9C9C9 sur fond #FFFFFF avec un ratio de 1,7.</p>
---	--

Correction

- Modifier la couleur pour obtenir un rapport de contraste de 3:1, au moins.
- Autrement, proposer un mécanisme alternatif pour afficher une version suffisamment contrastée de la page.

4. Tableaux

Donner un [résumé](#) à chaque [tableau de données complexe](#), s'assurer que pour chaque [tableau de données ayant un titre](#), ce titre est pertinent, identifier clairement les cellules d'en-tête, utiliser un mécanisme pertinent pour lier les cellules de données aux [cellules d'en-tête](#). Pour chaque [tableau de mise en forme](#), veiller à sa bonne linéarisation.

4.1 Tableaux de mise en forme

L'utilisation de tableaux de mise en forme pour positionner des contenus peut poser des problèmes aux personnes aveugles qui naviguent avec un lecteur d'écran. Il faut s'assurer que :

- L'ordre des contenus est cohérent quand le tableau est linéarisé, c'est-à-dire lorsqu'il est lu dans l'ordre des cellules et des lignes ;
- S'assurer que la balise `<table>` possède un attribut `role="presentation"` qui permet de ne pas en restituer la structure ;
- S'assurer que le tableau n'utilise pas de balises ou d'attributs propres aux tableaux de données : `<caption>`, `<th>`, `<thead>`, `<tfoot>`, `<colgroup>`, `scope`, `headers`, `axis`, `role="rowheader"`, `role="columnheader"`.

Correction

- Supprimer tous les attributs ou balises propres aux tableaux de données.
- Vérifier qu'il n'existe pas d'autres tableaux de mise en forme qui nécessiteraient des corrections pour les rendre accessibles.

5. Liens

Utiliser des [intitulés](#) de [liens](#) explicites, grâce à des informations de [contexte](#) notamment.

Les intitulés de lien, s'ils ne sont pas suffisamment explicites, peuvent poser des problèmes aux personnes aveugles et déficientes visuelles qui ne voient pas la page ou ne voient qu'une partie de la page. D'autres utilisateurs peuvent également être impactés comme les personnes handicapées intellectuelles par exemple. Il faut s'assurer que :

- Tous les liens possèdent un intitulé qui peut être le contenu textuel du lien ou l'alternative de l'image (attribut `alt` par exemple) pour les images-liens;
- L'intitulé est suffisamment explicite directement par l'intitulé ou par le moyen d'une information de contexte comme :
 - Le titre `<hx>` précédent;
 - Ou le contenu du paragraphe `<p>` dans lequel le lien est présent;
 - Ou le contenu de la ligne dans lequel le lien est présent;
 - Ou le contenu de la balise `` dans lequel le lien est présent;

- Ou le contenu de l'attribut title sur le modèle `title="[intitulé du lien] information complémentaire"`;
 - Ou le contenu de la propriété aria-label sur le modèle `aria-label="[intitulé du lien] information complémentaire"`;
 - Ou un passage de texte lié avec la propriété aria-labelledby.
- Le nom visible est contenu dans le nom accessible du lien. Par exemple, si le lien possède une propriété `aria-label`, le texte visible du lien doit s'y retrouver : `L'article le plus lu`.

De plus, on veillera également à ce que les liens qui ont le même intitulé (et le même contexte) mènent vers la même destination. Lorsque des liens aux intitulés identiques mènent vers des destinations différentes, l'intitulé du lien est alors considéré comme non pertinent, puisqu'il ne permet pas de les distinguer.

Note : Lorsque les liens contiennent uniquement des éléments graphiques (images, icônes, etc.), l'utilisation d'un attribut `title` pour créer une infobulle au survol de la souris est possible, même s'il n'apporte aucune information complémentaire. La valeur de ce titre doit alors être strictement identique à l'intitulé.

Constat

	<p>Dans la page « Liste des annuaires » plusieurs liens possèdent un intitulé qui n'est pas suffisamment clair. Dans l'attribut <code><a></code> ne pas remettre l'url du lien mais plutôt un titre global pour expliquer vers quoi l'utilisateur va être redirigé.</p>
---	---

Correction

- Vérifier qu'il n'existe pas des liens qui nécessiteraient des corrections similaires.

6. Scripts

Donner si nécessaire à chaque [script](#) une [alternative](#) pertinente. Avertir ou permettre le contrôle des scripts qui initient un [changement de contexte](#). Rendre possible le contrôle de chaque code [script](#) au moins [par le clavier et par tout dispositif de pointage](#) et s'assurer de

leur [compatibilité avec les technologies d'assistance](#) notamment pour les [messages de statut](#).

L'accessibilité des composants et fonctionnalités développés avec JavaScript impacte tous les utilisateurs et représente pour certains d'entre eux un enjeu majeur comme pour les personnes aveugles qui utilisent un lecteur d'écran ou les personnes qui n'utilisent pas la souris.

Le recours à l'API ARIA pour rendre les composants et les fonctionnalités compatibles requiert également une attention particulière quant à l'utilisation de certains de ses usages et de ses propriétés.

Pour rendre les composants et les fonctionnalités développées avec JavaScript accessibles, il est nécessaire de vérifier que :

- Les boutons sont des éléments HTML `<button>` ou utilisent le motif de conception ARIA `button` ;
- Les boutons ont des intitulés pertinents ;
- Les composants utilisent correctement le motif de conception ARIA qui leur correspond ;
- Les attributs ARIA comme `aria-hidden`, `aria-label`, `aria-live` par exemple et les rôles `presentation` ou `application` sont utilisés correctement ;
- La gestion du focus est cohérente avec les fonctionnalités qui modifient, ajoutent, suppriment ou mettent à jour du contenu dans la page ;
- Tous les composants et les fonctionnalités sont utilisables au clavier.

Calcul du nom accessible

L'API ARIA est associée à un document décrivant la méthode qui doit être utilisée par les agents utilisateurs pour déterminer quel doit être le nom accessible d'un composant lorsque plusieurs méthodes sont présentes.

Le calcul est le suivant :

- Si une propriété `aria-labelledby` est utilisée alors c'est le texte lié qui est le nom accessible ;
- Sinon si une propriété `aria-label` est utilisée c'est la valeur de l'attribut qui est le nom accessible ;
- Sinon c'est la méthode native (`alt`, `value`, contenu textuel) qui est le nom accessible ;
- Sinon c'est la valeur de l'attribut `title` qui est le nom accessible ;

- Lorsque l'attribut `title` est présent et qu'une autre méthode est également présente la valeur de l'attribut `title` devient alors une propriété de description qui s'ajoute au nom accessible.

Utilisation des modèles de conception ARIA

Lorsqu'un composant est développé sans recourir à des éléments HTML natifs, prévus à cet effet, il est essentiel qu'il respecte un motif de conception ARIA pertinent. Cela concerne par exemple, pour les plus communs : les fenêtres modales, les systèmes d'onglets, les zones qui s'affichent et se masquent, les datepickers, les listes en autocomplétions, les listes d'option, les carrousels, les sliders, les cases à cocher, les boutons radio, les spins buttons, les barres de progression.

Les modèles de conception ARIA permettent de définir la structure du composant, son rôle, ses propriétés et ses paramètres et, enfin, les interactions au clavier.

Ces modèles de conceptions sont décrits, avec des exemples réels, sur le site [ARIA Authoring Practices Guide](#).

6.1 Disclosure (Afficher / Cacher)

Le Disclosure est un élément qui va permettre de réduire (Cacher) ou de développer (Afficher) le contenu. Il comporte deux éléments : Un bouton pour le contrôle et une section de contenu qui sera la cible.

Constat

Des éléments contrôlés par des boutons pour afficher / cacher n'implémentent pas [le motif de conception Disclosure](#).

ACCÈS AU NUMÉRIQUE	AIDE ET SOUTIEN À LA PERTE D'AUTONOMIE	APPRENDRE ET SE FORMER	BUDGET - CONSOMMATION	DÉMARCHES - ACCÈS AUX DROITS - ACCOMPAGNEMENT	JUSTICE ET MÉDIATION
Information, orientation	Prévenir les risques liés à l'avancée en âge	Aides techniques et adaptation du logement au handicap et à la perte d'autonomie	Services de soins et de soutien à domicile	Téléassistance	Accueil de jour pour personnes âgées
					Transport - Mobilité
					Soutien aux aidants
					Hébergement temporaire
					Maisons de retraite et Etablissements pour personnes âgées dépendantes

Cette fonctionnalité n'est pas compatible avec les technologies d'assistance.



Correction

- Rendre les éléments utilisant le [modèle de conception Disclosure](#) totalement utilisable au clavier et notamment le fait de pouvoir cacher le composant avec la touche « echap ».
- Vérifier qu'il n'existe pas, dans les pages ou les écrans, d'autres zones qui nécessiteraient d'utiliser ou de compléter le motif de conception ARIA Disclosure.

6.2 Tab

Les Tabs sont un ensemble de sections de contenu superposées, qui affichent une section à la fois. Chaque section est associée à un onglet qui, une fois activé, affiche la section.

Correction

- Rendre les éléments utilisant le [modèle de conception Tabs](#) totalement utilisable au clavier et implémentant les rôles ARIA correctement.
- Vérifier qu'il n'existe pas, dans les pages ou les écrans, d'autres zones qui nécessiteraient d'utiliser ou de compléter le motif de conception ARIA Tabs.

6.3 Changements de contexte

Un changement de contexte est une situation où un utilisateur ne peut pas **anticiper le comportement d'une fonctionnalité** lorsque celle-ci ouvre une nouvelle page, valide un formulaire ou ajoute ou modifie du contenu dans la page par exemple.

Cela concerne plus spécifiquement les fonctionnalités qui se lancent sans que l'utilisateur puisse l'anticiper comme **la soumission automatique** d'un champ de formulaire sur la sélection d'un item ou lorsque l'utilisateur quitte un champ de saisie.

Correction

Une solution pour avertir l'utilisateur d'un changement dans une zone où les éléments sont variables est d'appliquer l'attribut `aria-live="polite"` sur la zone réactive.

7. Eléments obligatoires

Vérifier que dans chaque page Web, le code source généré respecte [les règles d'écriture](#) correspondant au [type de document](#), que le [titre](#) est pertinent et la [langue par défaut](#), indiquée. Vérifier que les balises ne sont pas utilisées [uniquement à des fins de présentation](#), que [les changements de langues](#) et de direction de [sens de lecture](#) sont indiqués.

7.1 Indication du type de document et validité du code HTML

Note : Depuis la parution dans le WCAG 2.2 de la suppression du critère 4.1.1 Analyse syntaxique, on peut considérer les critères 8.1 et 8.2 du RGAA comme systématiquement conforme.

7.2 Balises utilisées uniquement à des fins de présentation

Les éléments de structure HTML ont chacun une sémantique particulière (paragraphe, titre, image, lien, etc.). Si les éléments sont mal employés (détournés de leur utilité première), cela peut poser des problèmes aux utilisateurs qui naviguent à l'aide d'une technologie d'assistance (lecteur d'écran, plug-in...). En effet, les technologies d'assistance mettent des raccourcis à disposition permettant de naviguer rapidement entre certains types d'éléments (paragraphe, titres, listes, etc.). Si ces éléments sont mal employés, les utilisateurs ne peuvent pas utiliser ces fonctionnalités de repère et navigation dans le contenu.

Correction

- Supprimer les paragraphes vides, et réaliser les mises en forme (espacement) en utilisant uniquement CSS.
- Vérifier qu'il n'existe pas d'autres cas similaires.

8. Structuration de l'information

Utiliser des [titres](#), des [listes](#), et des citations pour structurer l'information. S'assurer que la structure du document est cohérente.

9. Présentation de l'information

Utiliser des [feuilles de styles](#) pour [présenter de l'information](#). S'assurer que l'information reste [compréhensible](#) lorsque les feuilles de styles sont désactivées. Vérifier l'effet de l'agrandissement à 200 % de la [taille des caractères](#) et de la redéfinition des propriétés d'espacement sur la lisibilité. S'assurer que les liens sont correctement identifiables, que la [prise de focus](#) est signalée et que l'utilisateur a le contrôle des contenus additionnels qui deviennent visibles au survol ou au focus. S'assurer que les [contenus cachés](#) sont ignorés par les technologies d'assistance et que l'information n'est pas donnée uniquement par la [forme, taille ou position](#) d'un élément.

9.1 Utilisation exclusive de CSS pour la présentation

Certains utilisateurs qui présentent des déficiences visuelles ou des troubles cognitifs (dyslexiques par exemple) vont avoir besoin d'adapter la présentation des pages avec leurs propres mises en forme. Cela est possible sans difficulté si le site web utilise exclusivement les feuilles de styles CSS pour réaliser les mises en forme. L'utilisation d'attributs et balises HTML de mise en forme rend ces adaptations plus compliquées, sinon impossibles.

[Le RGAA donne la liste des attributs et balises qu'il est interdit d'utiliser](#)

Correction

- Remplacer par des directives de styles CSS toutes les balises et tous les attributs de mise en forme référencés
- Vérifier qu'il n'existe pas, dans les pages ou les écrans, d'autres contenus qui ont recourt à des balises ou des attributs de mise en forme.

9.2 Visibilité du focus

- Les utilisateurs qui ne peuvent pas utiliser la souris ont besoin de pouvoir se **repérer dans la page** lorsqu'ils naviguent avec la touche de **tabulation**.
- Par défaut, le navigateur propose une **indication visuelle de la prise de focus** sous la forme d'un effet de bordure. Cet effet de bordure est personnalisable au moyen de la directive `outline`.



Supprimer cette indication visuelle de prise de focus, par exemple en utilisant des directives comme `outline: none`, rend la navigation dans le contenu très difficile et dans la plupart des cas impossible pour les personnes qui ne peuvent pas utiliser la souris.

Constat

	<p>Sur le bouton "Envoyer le commentaire" la prise de focus n'est pas suffisamment visible.</p>
	<p>Ici sur l'onglet é « Carte » le focus n'est pas visible.</p>
	<p>Le focus autour du violet n'est pas visible.</p>

Correction

- Surcharger les styles existants pour proposer une indication visuelle de la prise de focus personnalisée et dans ce cas, vous devez vous assurer que la prise de focus définie respecte les requis de contraste (**rapport de contraste de 3** minimum entre la couleur de l'indication du focus et les couleurs environnantes) ;
- Vérifier qu'il n'existe pas, dans les pages ou les écrans, d'autres éléments pour lesquels l'indication visuelle de la prise de focus serait également supprimée ou dégradée.

9.3 Information donnée par la forme, la taille ou la position

Lorsqu'une information est donnée uniquement par la forme, la taille ou la position (par exemple un texte agrandi, la position d'un symbole, une consigne qui indique un emplacement ou la forme d'un objet dans la page etc.) les personnes aveugles ne pourront pas avoir l'information.

Il est donc nécessaire de fournir cette information d'une autre manière, lorsque l'information fait partie d'un élément interactif, le plus simple est d'utiliser un `title`.

Note : cela ne concerne que les informations données par la forme, la taille ou la position d'élément ou de composant qui ne font pas partie d'un motif de conception ARIA. Par

exemple dans un système d'onglets, l'effet de mise en forme pour indiquer l'onglet actif est traité par l'utilisation d'une propriété spécifique.

Constat

	<p>Au sein de la pagination, la page active est signalé uniquement par sa couleur et sa forme.</p>
---	--

Correction

- Ajouter un `aria-current="true"` sur le lien de la page active ;
- Vérifier qu'il n'existe pas, dans les pages ou les écrans, d'autres indications données uniquement par la forme, la taille, ou la position.

10. Formulaires

Pour chaque formulaire, associer chacun de ses [champs](#) à son [étiquette](#), grouper [les champs de même nature](#) et leur donner une [légende](#), structurer les listes de choix de manière pertinente, donner à chaque [bouton](#) un intitulé explicite. Vérifier la présence de suggestions lors des erreurs de saisie, s'assurer que le [contrôle de saisie](#) est accessible, que la finalité des champs peut être déduite et que l'utilisateur peut garder le contrôle sur ses données à caractère financier, juridique ou personnel.

10.1 Contrôles de saisie et aide à la saisie

Les formulaires sont souvent accompagnés de message d'aide, d'erreur ou de confirmation. Certains utilisateurs comme les personnes aveugles peuvent ne pas prendre connaissance de ces messages. D'autres comme les personnes handicapées intellectuelles peuvent ne pas les comprendre.

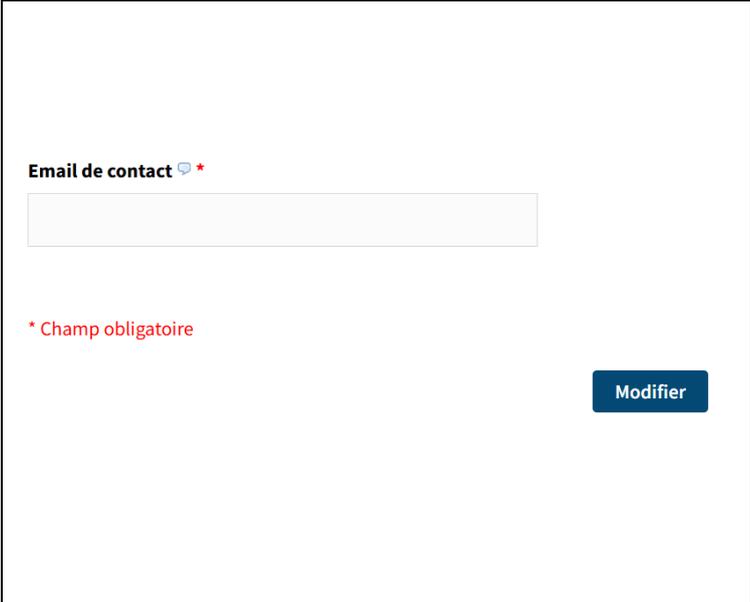
Plusieurs catégories de messages sont considérées : les indications de champs obligatoires, les messages d'erreurs de saisie, les indications de format obligatoires, les messages d'aides.



Les principes à respecter sont les suivants :

- Indiquer les champs obligatoires avant la saisie sauf si tous les champs optionnels sont indiqués. L'indication peut être un texte explicite situé avant le formulaire, dans le label, dans un texte lié au champ par la propriété `aria-describedby` ou par l'intermédiaire d'un attribut `title`. Cette indication peut être accompagnée ou non de la propriété `aria-required` ou l'attribut `required`
- Indiquer si nécessaire les formats de saisie obligatoires qui font l'objet d'une vérification. L'indication peut être dans le label, dans un texte explicite lié au champ par la propriété `aria-describedby` ou par l'intermédiaire d'un attribut `title`
- Lier tous les messages d'aide ou d'erreur aux champs concernés en positionnant le message d'erreur dans le label ou en le liant avec la propriété `aria-describedby`
- Ajouter dans les messages d'erreur liés aux formats obligatoires un exemple de saisie réelle, par exemple pour l'email, le téléphone, etc.

Constat

 <p>Email de contact *</p> <p>* Champ obligatoire</p> <p>Modifier</p>	<p>Le contrôle de saisie n'est pas pertinent.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> - Les champs obligatoires sont mal identifiés. <p>Corrections :</p> <ul style="list-style-type: none"> - Déplacer le texte "Champs obligatoires" juste avant le formulaire sur le modèle et le modifier par "les champs précédés d'un * sont obligatoires".
---	---

Correction

Ajouter un exemple de saisie réelle d'email dans le message d'erreur. Ces infobulles sont personnalisables avec [l'API de validation](#).

10.2 Indications de la nature des saisies (autocomplete)

La saisie d'un formulaire peut être particulièrement laborieuse et nécessiter des charges de travail considérables pour certains utilisateurs qui vont utiliser des technologies



d'assistance très complexes ou qui ne sont pas capables de comprendre les types de données attendues.

La WAI propose d'utiliser un système de déclaration de type de donnée normée au moyen de l'attribut `autocomplete` : par exemple `autocomplete="email"` indique qu'il s'agit d'une saisie d'adresse mail.

Ces indications peuvent être utilisées par le navigateur pour proposer des fonctionnalités de remplissage automatique des champs ainsi identifiés. Ce dispositif peut être d'une aide considérable pour les utilisateurs. Cela concerne plus spécifiquement les données à caractère personnel.

La liste des valeurs à utiliser est disponible ici : [Input Purposes for User Interface Components](#)

Note : il peut exister des situations, notamment liées à la sécurité, dans lesquelles ce dispositif pourrait ne pas être utilisé. Dans ce cas une dérogation pourrait être envisagée.

Constat

<p>Adresse</p> <input type="text" value="Adresse"/> <p>Comment s'y rendre</p> <input type="text" value="S'y rendre"/> <p>Email</p> <input type="text"/> <p>Téléphone</p> <input type="text"/>	<p>Au moins un champ qui attend une donnée personnelle de l'utilisateur n'est pas identifié.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> - Le champ "e-mail" - Le champ "téléphone" - Le champ « Adresse »
--	---

Correction

- Pour tous les champs référencés ci-dessous, implémenter un attribut `autocomplete` avec les valeurs correspondantes:
 - Page « Modifier mes coordonnées » :
 - Pour le champ « E-mail », utiliser `autocomplete="email"`.
 - Pour le champ « Téléphone », utiliser `autocomplete="tel"`.

- Pour le champ « Adresse », utiliser `autocomplete="address-level2"`.
- Vérifier qu'il n'existe pas, dans la page ou dans l'écran, d'autres champs qui nécessiteraient des adaptations similaires.

11. Navigation

Faciliter la navigation dans un [ensemble de pages](#) par au moins deux [systèmes de navigation](#) différents ([menu de navigation](#), [plan du site](#) ou [moteur de recherche](#)), un fil d'Ariane et l'indication de la page active dans le menu de navigation. Identifier les groupes de liens importants et la [zone de contenu](#) et donner la possibilité de les [éviter par des liens de navigation interne](#). S'assurer que [l'ordre de tabulation](#) est cohérent et que la page ne comporte pas de piège au clavier. S'assurer que les [raccourcis clavier](#) qui utilisent une seule touche sont contrôlables par l'utilisateur.

11.1 Landmarks

Si une structure HTML5 de document est présente, les landmarks viennent naturellement s'implémenter sur les balises correspondantes (hormis pour la recherche qui n'a pas d'élément HTML dédié) :

```
<header role="banner">
<div role="search">
<nav role="navigation">
<main role="main">
<footer role="contentinfo">
```

Si ce n'est pas le cas, les landmarks rôles peuvent venir s'implémenter sur les éléments englobant chaque zone :

```
<div role="banner">
<div role="search">
<ul role="navigation">
<div role="main">
<div role="contentinfo">
```

Exception faite du `role="navigation"` que l'on doit trouver sur tous les éléments de navigation de chaque page, **tous les rôles doivent être uniques dans la page.**

Correction

- Ajouter un `role="navigation"` sur la navigation principale.

11.2 Ordre de tabulation

La **navigation dans les contenus** peut être considérablement compliquée pour les personnes aveugles ou les personnes handicapées motrices qui naviguent au clavier si **l'ordre de tabulation** n'est pas **cohérent**.

L'ordre de tabulation est déterminé par l'ordre des éléments dans le code HTML. Il ne suit pas forcément l'ordre de lecture de la page ou de l'écran, mais il doit être cohérent en fonction de la nature des contenus et des fonctionnalités.

Lorsqu'il s'agit d'un composant qui correspond à un motif de conception ARIA comme un système d'onglets ou une fenêtre modale par exemple, l'ordre de tabulation est déterminé par le modèle lui-même.

Correction

- Modifier l'ordre des contenus dans le code HTML pour créer un parcours de tabulation cohérent
- Vérifier qu'il n'existe pas d'autres incohérences dans le parcours de tabulation des contenus et des fonctionnalités.

12. Consultation

S'assurer que l'utilisateur a le contrôle des actions imposées après un certain délai notamment les [procédés de rafraîchissement](#), Donner la possibilité de contrôler les [changements brusques de luminosité](#), les ouvertures de nouvelles fenêtres et les [contenus en mouvement ou clignotants](#). S'assurer que les expressions inhabituelles et le jargon sont explicités. Proposer des [versions accessibles](#) des documents en téléchargement ou les rendre accessibles. S'assurer que le contenu puisse être consulté quelque-soit l'orientation de l'écran et au moyen de [gestes simples](#). Permettre d'annuler les actions déclenchées par un mouvement et d'accéder aux mêmes fonctionnalités par une alternative, sans mouvement.