



Audit d'accessibilité simplifié - gimb.public.lu

Date : 10 août 2021

Appréciation générale

Niveau d'accessibilité global pour les critères testés : **bon**.

(Échelle : très faible, faible, moyen, bon, très bon)

Avertissement

Attention, l'audit effectué est un audit simplifié et non un audit de conformité (ou audit "complet"). Il a vocation à détecter une série de problèmes d'accessibilité mais n'est pas exhaustif. Le fait qu'aucun problème ne soit remonté pour un critère d'accessibilité donné ne signifie pas qu'il n'y a pas de problème d'accessibilité pour ce critère. De même, lorsque nous rapportons une occurrence d'un problème, ce problème peut avoir d'autres occurrences. Il est nécessaire de vérifier de manière exhaustive l'accessibilité de ce site conformément au référentiel [RGAA](#).

Échantillon de pages et référentiel

Voici les pages qui ont été évaluées lors de cet audit :

- Page P01 : [Gesond iessen - Méi bewegen - Luxembourg](#)
- Page P02 : [Contact — Gesond iessen - Méi bewegen - Luxembourg](#)
- Page P03 : [Actualités - Gesond iessen - Méi bewegen - Luxembourg](#)

Méthode d'évaluation : [Méthode de contrôle simplifiée de l'accessibilité pour le Luxembourg – v1.2](#)

Référentiel : [RGAA v4.1](#)

Documents bureautiques en téléchargement

L'accessibilité d'un échantillon de 69 fichiers bureautiques en téléchargement sur le site considéré a été analysée. Le niveau d'accessibilité global présenté dans la section "Appréciation générale", doit être nuancé au vu des points suivants :

Un nombre important de contenus dérogés impacte la pertinence des résultats de l'audit (la liste des dérogations prévues par la loi est disponible sur [la page "obligations" du site accessibilite.lu](#)). Le site contient un grand nombre de contenus dérogés. En effet, un nombre important de fichiers PDF a été publié avant le 23 septembre 2018 (soit 50 fichiers). Bien que ces fichiers soient exemptés d'obligation de mise en accessibilité par la loi du 28 mai 2019, il est recommandé de les rendre accessibles, sans quoi une part importante du contenu mis à disposition sur ce site ne pourra être consulté par les personnes en situation de handicap.



Pour information, le SIP met à disposition une liste de [critères d'accessibilité pour les documents bureautiques en téléchargement](#). Pour chaque document bureautique en téléchargement, il est possible de le rendre accessible en respectant les critères mentionnés précédemment, ou de proposer une alternative accessible, sous la forme d'une page Web ou d'un document Office (.docx, .pptx, etc.) proposant les mêmes informations.

Annexe technique

Thématique "images"

Recommandations générales

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. Pour trouver la bonne alternative textuelle pour une image donnée, vous pouvez vous aider de l'[arbre de décision proposé par la WAI](#).

Cas rencontré : images de décoration

Les [images de décoration](#) ne doivent pas avoir d'alternative textuelle et doivent être ignorées correctement par les technologies d'assistance. Dans le cas contraire, elles peuvent causer des problèmes de compréhension aux utilisateurs de lecteurs d'écran (aveugles et grands malvoyants). Pour une image *bitmap*, son attribut `alt` doit être vide. Pour une image vectorielle SVG, celle-ci doit avoir l'attribut `aria-hidden="true"`.

Exemples de problèmes détectés sur la [page P01](#)

Le lien riche vers l'article "le nutriscore, c'est quoi ?" contient une image et du texte. L'image est ici décorative, elle doit donc avoir une alternative textuelle vide. Actuellement son alternative est "Nutri-score".

Thématique "couleurs"

Recommandations générales

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.

Cas rencontré : informations données uniquement par la couleur

Ce type d'information est un problème pour les personnes déficientes visuelles, aveugles ou par exemple les personnes qui ne voient pas certaines couleurs ou ne perçoivent simplement pas les couleurs. Pour chaque information véhiculée par la couleur, il est nécessaire de mettre en place une alternative, comme par exemple un changement de style (graisse du texte, taille du texte, soulignement, etc.)



Exemples de problèmes détectés sur la [page P01](#)

Dans le menu de navigation, l'information de l'élément actif est donnée uniquement par la couleur. Il est nécessaire de donner cette information par d'autres moyens, notamment via une modification du style (taille du texte, grasse, soulignement, bordure, ...) ainsi que l'attribut `aria-current="page"`.

Cas rencontré : contrastes des textes

Les contrastes de couleurs sont importants pour plusieurs types de déficiences visuelles comme celles des grands malvoyants ou des personnes ayant des problèmes de perception des couleurs. Les contrastes minimaux d'un texte sur le fond de page sont définis par le RGAA comme suit :

- Pour un texte sans effet de grasse
 - De taille inférieure à 24 px : le contraste minimum est de 4.5:1 ;
 - De taille supérieure ou égale à 24 px : le contraste minimum est de 3:1.
- Pour un texte en gras
 - De taille inférieure à 18.5 px : le contraste minimum est de 4.5:1 ;
 - De taille supérieure à 18.5 px : le contraste minimum est de 3:1.

Pour vérifier les contrastes, on peut utiliser un outil tel que [« Colour Contrast Analyser »](#). Si les problèmes de contraste ne peuvent être résolus simplement, il est aussi possible d'utiliser un *« style switcher »*. C'est une fonctionnalité proposée par le site qui permet de renforcer les contrastes pour les personnes qui ont des problèmes avec les couleurs. Un exemple de *« style switcher »* est disponible sur le site [snf.com](#), dans son menu *« Accessibilité »*.

Exemples de problèmes détectés sur la [page P01](#)

Certains textes sur cette page n'ont pas un contraste suffisant (p.ex le texte de la langue en cours "fr").

Thématique "liens"

Recommandations générales

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

Cas rencontré : pertinence des intitulés

Chacun doit pouvoir comprendre aisément la fonction et la destination de chaque lien. Les problèmes rencontrés ici le sont pour les aveugles, les malvoyants, les handicapés moteurs qui naviguent à la voix et les handicapés cognitifs.

Exemples de problèmes détectés sur la [page P01](#)

Dans le carrousel, le lien-image "journée internationale des fruits et légumes 2021" a un intitulé qui ne reprend pas son contenu visible (ex: les fruits et légumes, des éléments essentiels de ton alimentation). Cela peut poser problème pour les utilisateurs de logiciels de reconnaissance vocale.



Thématique "scripts"

Recommandations générales

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](#).

Cas rencontré : éléments interactifs inaccessibles au clavier

Les composants riches développés en JavaScript doivent respecter des modèles de conception spécifiques pour pouvoir être considérés comme accessibles (modèles de conception décrits dans le document [« WAI-ARIA Authoring practices »](#)). Sans cela le composant ne sera pas correctement restitué aux utilisateurs de lecteurs d'écran qui ne sauront pas comment l'utiliser. Ces composants doivent notamment utiliser des interactions au clavier spécifiques, sans lesquelles ils seront inutilisables pour les utilisateurs de la navigation au clavier.

Exemples de problèmes détectés sur la [page P01](#)

Le bouton de recherche a un `role="button"` mais n'est pas activable via la barre d'espace. Il ne respecte pas le design pattern "button" <https://w3c.github.io/aria-practices/#button> C'est aussi le cas du bouton du menu mobile. On ne peut atteindre les sous-menus du menu de navigation au clavier. Le menu ne respecte pas le design pattern "Disclosure" <https://w3c.github.io/aria-practices/#disclosure> <https://w3c.github.io/aria-practices/examples/disclosure/disclosure-navigation-hybrid.html> Le carrousel ne respecte pas le design pattern "carousel" <https://w3c.github.io/aria-practices/#carousel> En effet, les petits ronds qui permettent d'indiquer la position doivent respecter le pattern "tabs" <https://w3c.github.io/aria-practices/#tabpanel> on doit pouvoir se déplacer dans ces éléments avec les flèches gauche et droite du clavier.

Thématique "éléments obligatoires"

Recommandations générales

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.

Cas rencontré : validité du code source de la page

Les lecteurs d'écran s'appuient sur la sémantique des balises HTML pour restituer le contenu et naviguer, si la syntaxe HTML n'est pas respectée ou si les balises sont détournées, cela peut avoir un impact sur la navigation qui peut devenir inopérante et la restitution qui peut devenir incompréhensible. Pour vérifier la validité de la syntaxe HTML, on peut utiliser le [validateur HTML du W3C](#) avec le *bookmarklet* [« WCAG 2.1 parsing error »](#). Ce *bookmarklet* filtre les résultats et ne montre que les erreurs ayant un impact sur l'accessibilité.



Exemples de problèmes détectés sur la [page P01](#)

Vérifier que les listes sont structurées correctement

Documentation : <https://dequeuniversity.com/rules/axe/4.3/list?application=webdriverjs&lang=fr>

Occurrences :

- Corriger tous les éléments suivants : L'élément de liste comporte des descendants directs avec un rôle qui n'est pas autorisé : `tabpanel` (Ex : `.js-slideshow-ribbon`)
- Corriger tous les éléments suivants : L'élément de liste comporte des descendants directs qui ne sont pas autorisés à l'intérieur de l'élément `` (Ex : `.last-publication > ol`)

Vérifiez que le code html de la page est valide

Documentation : <https://validator.w3.org/nv/>

Occurrences :

- L'élément "ol" n'est pas autorisé comme enfant de l'élément "ol" dans ce contexte. (Ex : `<ol class="search-results">`)

Cas rencontré : indication de langue

Les lecteurs d'écran utilisent les indications de langue pour vocaliser correctement le contenu. La langue principale de la page est spécifiée via l'attribut `lang` sur l'élément `<html>` . Lorsqu'un mot d'origine étrangère est inséré dans du contenu écrit dans la langue principale de la page, il doit posséder si nécessaire une indication de langue. L'indication de langue se fait par l'intermédiaire de l'attribut `lang` . Il existe néanmoins des exceptions :

- Lorsqu'il s'agit d'un nom, l'indication de langue doit être faite uniquement quand le nom doit se prononcer dans sa langue d'origine ;
- Lorsqu'il s'agit d'un mot d'origine étrangère, présent dans le dictionnaire de la langue principale de la page, l'indication de langue n'est pas nécessaire ;
- Lorsqu'il s'agit d'un mot d'origine étrangère d'usage courant, mais absent du dictionnaire, l'indication de langue doit être faite uniquement si la prononciation dans la langue principale de la page est problématique.

Exemples de problèmes détectés sur la [page P01](#)

Cette page contient des textes en d'autres langues non marqués par un changement de langue (p.ex. "Gesond iessen")



Thématique "présentation de l'information"

Recommandations générales

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.

Cas rencontré : contenus cachés

Tout contenu dans la page peut être ignoré par les technologies d'assistance (notamment en utilisant l'attribut `aria-hidden="true"`). Si un contenu est visible mais ignoré par les technologies d'assistance, les utilisateurs de ces technologies d'assistance ne pourront pas prendre connaissance du contenu ni interagir avec les éventuels éléments interactifs qu'ils pourraient contenir. La visibilité d'un contenu doit donc être cohérente pour tous les utilisateurs du site, qu'ils utilisent ou non une technologie d'assistance.

Exemples de problèmes détectés sur la [page P01](#)

les flèches permettant de se déplacer dans le carrousel sont dans un conteneur qui a l'attribut `aria-hidden="true"`. Ceci est un problème car les boutons sont focalisables, ils peuvent être atteints par un lecteur d'écran qui ne pourra pas restituer leur intitulé à cause du `aria-hidden="true"`. Il faut soit les rendre non-focalisables via `tabindex="-1"` soit retirer le `aria-hidden="true"`.

Thématique "formulaires"

Recommandations générales

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.

Cas rencontré : identification des champs, des contrôles et des regroupements de formulaires

L'identification de ces éléments fournit aux aveugles et grands malvoyants les informations nécessaires pour pouvoir remplir un formulaire. Les handicapés moteurs utilisant un système de reconnaissance vocale s'appuient aussi sur ces éléments pour se déplacer dans les différents [champs](#) et actionner les boutons. Pour associer une [étiquette](#) (`<label>`) à un champ de formulaire on peut utiliser l'attribut `for` de l'étiquette dont la valeur doit être identique à celle de l'attribut `id` du champ. Les [champs de même nature](#) doivent être regroupés, cela peut être réalisé via l'élément `<fieldset>` auquel on doit fournir une [légende](#) via l'élément `<legend>`.



Exemples de problèmes détectés sur la [page P01](#)

L'étiquette du champ de recherche n'est pas visible. Un placeholder n'est ici pas suffisant, car il ne sera plus visible une fois que l'utilisateur aura commencé à remplir le champ. Une solution peut être de placer un attribut `title` sur le champ, qui contient le contenu de l'étiquette. Ce contenu sera visible au hover, même pendant la saisie.

Cas rencontré : contrôle et aide à la saisie

Que ce soit pour les handicapés cognitifs ou pour les aveugles et déficients visuels, il est nécessaire d'explicitier les formats de données attendus dans les consignes et les messages d'erreur, ainsi que le caractère obligatoire de la saisie de certains champs.

Exemples de problèmes détectés sur la [page P02](#)

Le type de données attendu dans le champ e-mail n'est pas précisé, ni dans l'étiquette, ni dans le message d'erreur. Il est ici nécessaire de donner un exemple.

Ce rapport est publié par le [service information et presse](#) sous [licence CC0](#).