



Audit d'accessibilité simplifié - bettendorf.lu

Date : 12 juillet 2024

Appréciation générale

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

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

Avertissement

Il s'agit là d'un audit simplifié et non d'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 [RAWeb](#).

Échantillon de pages et référentiel

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

- Page 1 : [Accueil - Bettendorf](#)
- Page 2 : [Contactez-nous - Bettendorf](#)
- Page 3 : [La commune en chiffres - Bettendorf](#)

Méthode d'évaluation : [Méthode de contrôle simplifiée de l'accessibilité pour le Luxembourg – v1.2.1](#)

Référentiel : [RAWeb v1](#)

Déclaration sur l'accessibilité

La déclaration sur l'accessibilité est incomplète. Certaines informations obligatoires sont manquantes, par exemple : niveau de conformité, mécanisme de retour d'information, liens vers les sites du SIP et du médiateur. Les éléments obligatoires de la déclaration sont présentés dans l'article 5 de la [loi du 28 mai 2019](#). Une fois la déclaration d'accessibilité mise à jour, l'éditeur du site a 30 jours pour en informer le SIP par mail à l'adresse accessibilite@sip.etat.lu.

Documents bureautiques en téléchargement

L'accessibilité d'un échantillon de 128 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 :

- Une quantité importante de documents bureautiques en téléchargement a été détectée, ce qui peut avoir un impact négatif sur l'accessibilité globale de ce site. Sur l'échantillon de fichiers analysé, 31 % sont des documents PDF dont 2 % de formulaires PDF. Le format PDF est en général moins accessible que les pages Web et que les documents Office (.docx, .pptx, etc.). L'accessibilité des formulaires est particulièrement importante dans la mesure où ils sont en général nécessaires à la réalisation de procédures administratives.
- Sur les fichiers PDF qui entrent dans le cadre de la loi (publication après le 23 septembre 2018) et issus de l'échantillon, 57 % ont un problème d'accessibilité grave, qui empêche totalement l'accès au contenu du document par les utilisateurs (ex : fichier non balisé, fichier protégé, document numérisé). Ces informations sont données à titre indicatif, car la présence d'une alternative accessible n'a pas été vérifiée dans le cadre de cet audit simplifié.

Pour information, le SIP met à disposition le [référentiel d'évaluation de l'accessibilité des documents au format PDF \(RAPDF\)](#). Pour chaque document PDF en téléchargement, il est possible de le rendre accessible en respectant les critères de ce référentiel, 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 porteuses d'information

Les images porteuses d'information doivent avoir une alternative textuelle qui sera restituée aux personnes utilisant un lecteur d'écran, aveugles et grands malvoyants. Cette alternative textuelle doit fournir l'information véhiculée par l'image, il ne s'agit pas d'une description de l'image. Pour une image matricielle, son attribut `alt` doit contenir cette alternative textuelle. Pour une image vectorielle SVG, celle-ci doit avoir l'attribut `role="img"` et son alternative textuelle pourra être fournie via les attributs `aria-label` ou `aria-labelledby`.

Exemples de problèmes détectés sur la [page 1](#)

Certaines images, comme par exemple "Club Senior Nordstad" dans le carrousel des partenaires, n'ont pas d'alternative textuelle.



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 matricielle, 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 1](#)

Certaines images, comme par exemple le blason de la commune en pied de page, doivent bénéficier d'un attribut `alt` vide (ici il est précisé "logo-footer").

Cas rencontré : images porteuses d'information complexes

Certaines images porteuses d'informations sont particulièrement complexes, par exemple lorsqu'elles présentent des graphiques, schémas, cartes, etc. Dans ce cas, ces images nécessitent une description détaillée adjacente à l'image afin de permettre aux personnes aveugles et dyslexiques d'accéder à toutes les informations véhiculées par l'image. Pour mettre à disposition cette description détaillée, plusieurs solutions sont possibles. Le texte peut être adjacent à l'image dans la même page ou on peut placer un lien adjacent à l'image qui mène à une page où se trouve la description. Le texte peut aussi être inclus dans un composant permettant de le masquer et de l'afficher sur demande, au clic d'un bouton « afficher la description de l'image » (via le motif de conception ARIA « [disclosure](#) »).

Exemples de problèmes détectés sur la [page 3](#)

Les trois diagrammes ne sont pas décrits de manière détaillée dans un bloc de texte adjacent. La description détaillée d'une image porteuse d'information complexe ne doit pas être fournie dans l'attribut `alt` de l'image, mais elle doit être présente dans un bloc de texte adjacent à l'image et de manière structurée via l'utilisation de titres, de listes, etc. Toute l'information véhiculée par l'image complexe doit être présente dans la description détaillée. Lorsqu'une description détaillée est présente, l'alternative textuelle de l'image (attribut `alt`) doit contenir le titre de cette image et une référence à la description détaillée (par exemple : "voir la description détaillée de cette image ci-dessous").

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 : 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 RAWeb 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 sélecteur de styles. 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 sélecteur de styles est disponible sur le site guichet.lu, dans son menu « Accessibilité ».

Exemples de problèmes détectés sur la [page 1](#)

"Accessibility tools" crée de nouveaux problèmes de contraste qui n'existent pas dans la version "standard" : avec le réglage "High Contrast", les titres sous "Actualités" deviennent illisibles, contraste de 1.03:1 contre 3:1 attendu comme minimum requis. D'autres intitulés ou liens n'ont pas le rapport de contraste suffisant.

Thématique "liens"

Recommandations générales

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



Cas rencontré : présence d'intitulés

Un lien doit toujours avoir un intitulé. Celui-ci permet aux utilisateurs de comprendre la destination du lien. La présence d'intitulés sur tous les liens est particulièrement importante pour les aveugles et malvoyants. Si un lien n'a pas d'intitulé, les lecteurs d'écran vont lire simplement « lien » sans annoncer sa destination.

- Pour un lien qui ne contient qu'une image, l'intitulé du lien est l'alternative de l'image. Pour une image matricielle, son attribut `alt` doit contenir l'intitulé du lien (ex : ``). Pour une image vectorielle SVG, celle-ci doit avoir l'attribut `role="img"` et l'intitulé du lien pourra être fourni via les attributs `aria-label` ou `aria-labelledby` (ex : `<svg role="img" aria-label="facebook">...</svg>`).
- Lorsqu'un lien contient du texte et des images, l'intitulé du lien correspond à la concaténation des intitulés de ces différents éléments contenus dans le lien (ex : pour ` Facebook` l'intitulé du lien est « logo facebook facebook ». Dans cet exemple, cet intitulé n'est pas pertinent et l'alternative de l'image doit être vide.)

Exemples de problèmes détectés sur la [page 1](#)

Plusieurs liens-images comme le blason de la commune (Accueil), les logos Facebook et RSS ou les logos des partenaires n'ont pas d'alternative textuelle.

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 guide »](#)). 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 1](#)

Le menu burger ou le carrousel ne sont pas accessibles au clavier. Pensez à respecter les motifs de conception ARIA pour chacun de ces domaines. Le menu d'accessibilité n'implémente pas correctement le motif de conception ARIA [Dialog](#).



Thématique "éléments obligatoires"

Recommandations générales

Vérifier que dans chaque page Web, le [titre](#) est pertinent et la [langue par défaut](#) est 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 : 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.

Plus d'information sur le sujet dans notre article : [L'attribut lang, pour rendre la page audible... et compréhensible](#)

Exemples de problèmes détectés sur la [page 1](#)

Cette page contient des textes en d'autres langues non marqués par un changement de langue via l'attribut `lang` (par exemple « Jugendtreff Nordstadjugend »).

Thématique "structuration de l'information"

Recommandations générales

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

Cas rencontré : structure du document HTML5

La structuration du document HTML5 permet aux aveugles, grands malvoyants et handicapés moteurs de naviguer très rapidement entre les zones principales de la page (header, footer, zone de contenu principale, navigation, ...)

Exemples de problèmes détectés sur la [page 1](#)

La zone de contenu principal n'est pas structurée par une balise html5 `<main>`.



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 : visibilité du focus

Les handicapés moteurs qui naviguent au clavier utilisent l'[indicateur de focus](#) fourni par le site sur les éléments interactifs pour savoir où ils se situent dans la page. L'indicateur de focus se déplace via les touches `tab` et `shift-tab`. L'indicateur de focus par défaut peut être désactivé via CSS, dans ce cas il est nécessaire de changer le style de l'élément interactif pour rendre l'indicateur de focus visible (sa couleur devra avoir un contraste minimum de 3:1 avec l'arrière-plan contigu).

Exemples de problèmes détectés sur la [page 1](#)

Dans de nombreux éléments de cette page, le focus n'est pas visible.

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 1](#)

Les puces du carrousel sont destinataires d'un attribut `aria-hidden`, ce qui les rend inaccessibles aux technologies d'assistance.

Cas rencontré : information donnée par la forme, la taille ou la position

Ce type d'information est un problème pour les personnes aveugles et grands malvoyants. Pour chaque information véhiculée par la forme, la taille ou la position, il est nécessaire de mettre en place une alternative.

Exemples de problèmes détectés sur la [page 1](#)

Un exemple : les puces du carrousel. La puce active bénéficie bien d'un changement de style (class `slick-active`) mais l'attribut `aria-selected` reste figé à `false`.



Cas rencontré : contenus additionnels au survol et au focus

Les utilisateurs doivent pouvoir garder le contrôle des contenus additionnels qui apparaissent au survol et au focus (ex : infobulles, menus déroulants). Tout élément qui apparaît au survol doit aussi pouvoir apparaître au clavier, lorsque l'élément prend le focus. Pour les malvoyants qui utilisent une loupe d'écran, ces contenus apparaissant au survol peuvent perturber la consultation du site. Ils doivent pouvoir être masqués simplement. Si le contenu apparaît hors de la zone affichée par la loupe, il doit pouvoir être survolé à la souris.

Exemples de problèmes détectés sur la [page 1](#)

Les sous-menus du menu de navigation apparaissent via des styles CSS uniquement, ils ne peuvent être rendus visibles au clavier.

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 1](#)

Le champ de recherche ne dispose pas d'étiquette (l'étiquette est bien présente mais dispose d'un attribut CSS `display: none` qui risque d'en empêcher sa restitution par les technologies d'assistance). 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 2](#)

Le type de données attendu dans le champ « e-mail » n'est pas précisé dans le message d'erreur. Il est nécessaire de donner un exemple. Par ailleurs il convient d'expliciter en aval du formulaire la signification de l'astérisque.

Thématique "navigation"

Recommandations générales

Proposer au moins deux [systèmes de navigation](#) différents dans un [ensemble de pages](#) ([menu de navigation](#), [plan du site](#) ou [moteur de recherche](#)). Donner la possibilité d'éviter ou d'atteindre les principaux regroupements de contenus en particulier la zone de contenu principale via un [lien d'évitement ou d'accès rapide](#). 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](#) n'utilisant qu'une seule touche sont contrôlables par l'utilisateur.

Cas rencontré : *landmarks* ARIA

Les utilisateurs aveugles utilisent pour naviguer rapidement dans une page des points de repères ou [landmarks](#). Ceux-ci définissent les principales zones de la page comme l'entête, le menu de navigation, la zone de contenu principale, le pied de page, le moteur de recherche. Chacune de ces zones doit avoir un attribut `role` dont la valeur correspond au type de zone :

- `role=banner` pour l'entête,
- `role=navigation` pour le menu de navigation,
- `role=main` pour la zone de contenu principale,
- `role=contentinfo` pour le pied de page,
- `role=search` pour le moteur de recherche.

Exemples de problèmes détectés sur la [page 1](#)

Les zones d'entête, de contenu principal, de pied de page ne peuvent être atteintes ou évitées. Il est nécessaire de mettre sur ces zones un attribut `role`, avec la valeur appropriée correspondante : "banner", "main", "contentinfo".

Cas rencontré : liens d'accès rapide

Les [liens d'accès rapide](#) sont des liens présents en début de page et permettent aux utilisateurs qui naviguent au clavier et aux utilisateurs malvoyants qui utilisent une loupe d'écran d'éviter des zones de contenus redondants comme l'entête et la navigation. Il est indispensable d'avoir au moins un lien d'accès rapide vers la zone de contenu principale (`<main>`). Ces liens peuvent être positionnés hors écran et apparaître à la prise de focus.

Exemples de problèmes détectés sur la [page 1](#)

Le lien "Aller au contenu principal" n'a aucun effet.



Thématique "consultation"

Recommandations générales

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 quelle que 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.

Cas rencontré : contenus en mouvement ou clignotants

Ces contenus posent problème aux utilisateurs avec des difficultés de lecture ou des troubles de l'attention. Les utilisateurs doivent avoir la possibilité de mettre le mouvement en pause, ou de masquer le contenu en mouvement.

Exemples de problèmes détectés sur la [page 1](#)

La page contient un carrousel qui ne peut pas être arrêté. La présence d'un bouton « pause » est ici indispensable.

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