

# Audit accessibilité 2022

-- Mail Alexandre --

À titre d'information, il existe une liste éditée chaque année de tous les sites public luxembourgeois qui vont être audités pour l'accessibilité.

[https://data.public.lu/fr/datasets/echantillon-pour-le-controle-de-laccessibilite-numerique-2022/#\\_](https://data.public.lu/fr/datasets/echantillon-pour-le-controle-de-laccessibilite-numerique-2022/#_)

**Il y a 3 type d'audit :**

- Complet
- Simplifié
- Application mobile

**Dans les simplifiés, nous avons quelques site à nous:**

- bertrange.lu
- kehlen.lu
- mersch.lu
- steinfort.lu
- syvicol.lu
- valorlux.lu

**Déclaration:**

- <https://accessibilite.public.lu/fr/tools/decla.html>

Ne pas oublier de faire un mail à [accessibilite@sip.etat.lu](mailto:accessibilite@sip.etat.lu) en précisant le site et la page de déclaration

Ne pas oublier de faire une liste de contenu non accessible

## Sites:

- <https://accessibilite.public.lu/>

## Outils pour audit interne

### Bookmarks:

- BOSA Accessibility Check - <https://openfed.github.io/AccessibilityCheck/>
- Sa11y - <https://sa11y.netlify.app/bookmarklet/>

### Plugins:

- Arc Toolkit - <https://chrome.google.com/webstore/detail/arc-toolkit/chdkkkccnlfncngelccgbgfmjebmkmce?hl=en>
- Wave Web Accessibility - <https://wave.webaim.org/>
- Equalweb (Payant mais exist en gratuit) - <https://www.equalweb.com/>
- Axe DevTool - <https://chrome.google.com/webstore/detail/axe-devtools-web-accessib/lhdoppojpmngadmndnejefpokejbdd>
- Contrast checker:
- WCAG - Contrast checker - <https://contrastchecker.com/>
- Plugin contrast checker - <https://chrome.google.com/webstore/detail/wcag-color-contrast-check/plnahcmalebffmaghcpcmpaciebdhgdf>

## Best practices

- Mettre en place un bouton "Go to Content" et si possible "Go to sitemap"
- Outline au focus
- Select natif sur mobile
- Attribut 'alt' ou 'title' sur image et lien
- Attribut 'role' et aria label sur les éléments principaux
- Focus à l'ouverture d'une modal
- Ordre des balises titres
- Essayer d'obtenir que des contrastes 4.5:1 (note AA)
- Repère visuel sur les champs de formulaire pour montrer que l'on est dans un champs

- Balise label pour tous les champs de formulaires
- Message d'erreur indiquant le problème rencontré à la validation d'un champ
- Différencier les liens déjà visités
- Taille de la font et des icons
- S'assurer que le visiteur sache qu'une vidéo est jouée
- Faire attention à l'ordre de focus sur les sliders
- Préciser dans le titre quand un lien s'ouvre dans une nouvelle fenêtre
- Caption sur les tableaux pour expliquer le contenu
- Ajouter un bouton de contraste si pas possible de changer les couleurs du site

## Points importants qui reviennent dans les audit

### 1. Lien d'accès rapide

1. Lien pour accéder rapidement au contenu du site (bypass la nav, etc..)
2. Lien pour aller au sitemap
3. Autre raccourci qui pourrait être utile dans le cadre du site internet

### 2. Contraste des couleurs (texte/background)

1. Il est possible de jouer sur les nuances de couleurs, mais aussi sur la taille de la font.
2. Utiliser l'outil : <https://webaim.org/resources/contrastchecker/>. Il faut être **VERT**, **pas de rouge**.
3. Ajouter un bouton pour activer le contraste renforcé pour ne pas casser la corpo si besoin.

### 3. Balise ALT sur les images

1. Les balises **ALT** sur les images ne doivent être **remplies** que si le **contenu de l'image est pertinent** (il faut décrire au mieux le contenu de l'image dans ce cas présent)
2. Si une image n'a qu'une vocation de **décoration** ou de contenu graphique, **la balise ALT doit être VIDE**
3. Un arbre de décision est disponible ici pour aider :  
<https://www.w3.org/WAI/tutorials/images/decision-tree/>

### 4. Navigation par TAB

1. Le site doit être navigable via la touche tabulation du clavier
2. **Le focus doit donc être visible pour chaque élément**
3. Attention aux sliders, aux éléments masqués, aux librairies js et aux sous navigation (point suivant)
4. **On ne doit pas perdre la visibilité du focus et faire des TAB invisibles**
5. L'ordre doit être logique, on ne saute pas du header au footer pour retourner au main content, etc...

## 5. Naviguer par TAB (sous navigation)

1. la personne doit avoir le **choix d'ouvrir une sous navigation ou/et de fermer cette sous-navigation** (exemple: <https://www.kehlen.lu/>)
2. Eviter donc, de devoir passer par tous les points des sous-navigations pour continuer à avancer dans la page

## 6. Formulaires

1. Chaque input, doit être rattaché à un label qui indique ce qu'on attend pour ce champ
2. `<label for="inputId">blabla</label><input type="text" id="inputId">`
3. Idéalement, dans le cadre de valeurs avec un formatage, il est important de donner un exemple de formatage (dates, heures, email, ...)
4. Il est possible de faire des groupes de champs similaires avec le tag `<fieldset>` et d'indiquer une description via le tag `<legend>`

## 7. Balise title & actions claires

1. Ajouter une balise title sur les liens/buttons/actions, si le contenu de l'élément n'est pas clair
2. Cela permet à la personne de savoir ce qu'il se cache derrière le lien

## 8. Test via lecteur d'écran

1. Il est bon de tester rapidement le rendu du site avec un lecteur audio pour déceler les problèmes évidents
2. Avec MAC OS, aucune excuse. Il suffit de faire CMD + F5

## 9. Aria-current

1. Ajouter l'attribut '**aria-current="page"** sur les liens qui concernent la page actuelle dans les différentes navigation
2. Cela permet à la personne de savoir que le lien en focus est l'url actuelle

## 10. Librairie JS

1. Essayer la navigation via la touche TAB sur les composants JS
2. Adapter pour que cette navigation soit la plus facile possible et dans le cas contraire utiliser des librairies optimisées pour l'accessibilité

## 11. Code source - Validité W3C

1. Le code HTML doit être au maximum valide W3C. Utiliser :  
<https://validator.w3.org/nu/>
2. Utiliser le booklet qui permet de filtrer les erreurs/warning qui concernent l'accessibilité : <https://www.tpgi.com/wcag-2-0-parsing-error-bookmarklet/>

## 12. **Landmarks ARIA**

1. Ajouter les landmarks ARIA. 5 au total :
  1. role="banner" => header
  2. role="navigation" => pour les navigation
  3. role="main" => zone de contenu principale
  4. role="contentinfo" => pour le footer
  5. role="search" => pour le moteur de recherche

## 13. **Utiliser les ARIA**

1. Liste disponible ici : <https://www.w3.org/TR/html-aria/> (bonne chance)
2. Souvent les librairies / bootstrap remplissent déjà pas mal ce genre d'attributs (dropdown, accordion, ...)
3. Par exemple role="list" sur les ul / ol

## 14. **Pouvoir mettre pause sur les éléments qui défilent**

1. Il doit être possible de mettre pause sur les éléments de type highlight / marquee

## 15. **Ordre des titres**

1. Comme en SEO, il est important d'imbriquer correctement les titre H1, H2, H3, H4, H5 pour garder la compréhension et le rattachement des titres entre eux

## 16. **Langue différente**

1. Ajouter l'attribut lang sur la balise HTML
2. Si des textes ne sont pas dans la langue de la page, il est normalement conseillé de rajouter un attribut lang sur le texte en question