

Mémo Accessibilité Web

E-Accessibility Solutions for Everyone

Table des matières

- Contenu textuel
- Contenu non textuel
- Couleurs et contrastes
- Navigation clavier
- Navigation générale
- Mise en page
- Les attributs ARIA
- La boîte à outils



Recommandations Orange :

<https://a11y-guidelines.orange.com/fr/>

Contenu textuel

Donner un titre aux pages

Chaque page doit avoir un titre présentant son contenu.

```
<title> Titre de la page - nom du site </title>
```

Donner des titres aux rubriques

Vérifier la structure des pages et identifier les titres.

Utiliser des balises de titres **<h1>** à **<h6>**.

```
<h1> Accueil - Orange </h1>
```

```
<h2> Les actualités </h2>
```

Indiquer la langue principale de la page

```
<html lang="fr">
```

Indiquer les changements de langue

```
Découvrir Orange <span lang="en">live</span>
```

Respecter la sémantique HTML

- Les éléments doivent avoir des balises de début et fin.
- Les liens sont bien des liens et renvoient vers une page.
- Les boutons sont bien des boutons, ils déclenchent une action, une validation ou permettent un choix (**<button>** ou **<input type="button">**).
- Si besoin utiliser des listes HTML (****, **** et ****).
Pas de liste vide ou ne contenant qu'un seul élément.

Structurer les tableaux de données

- Ne pas utiliser les tableaux à des fins de mise en page. Utiliser les balises **<table>**, **<tr>**, **<th>** et **<td>**.
- Pour les tableaux complexes, utiliser l'attribut **scope** (**row/col**) et les attributs **headers** et **id**.
- Ajouter un résumé dans la balise **<caption>**.
- Valider le code HTML des pages produites à l'aide du validateur W3C : <https://validator.w3.org/>

Associer une étiquette aux champs de formulaire

À l'aide d'une balise **label** (attributs **for** et **id**) :

```
<label for="email">Adresse email</label>
```

```
<input id="email" type="text">
```

À l'aide d'un attribut **title** et d'un attribut **aria-label** :

```
<input type="text" title="terme à rechercher" aria-label="terme à rechercher">
```

Contenu non textuel

Fournir un équivalent textuel pour les images

Image porteuse d'information :

```

```

Image décorative :

```

```

Fournir une transcription pour les pistes audio/vidéo

Fournir une transcription textuelle dans la page ou via un lien vers un fichier texte des pistes audio ou vidéo.

Couleurs et contrastes

Vérifier le contraste des couleurs

- 4.5/1 min. pour du texte de taille standard.
- 3/1 min. pour texte de grande taille (150% du texte normal ou 120% si gras).

Vous pouvez utiliser l'application « Colour Contrast Analyser » : <https://www.tpgi.com/color-contrast-checker/>

Ne pas s'appuyer que sur la couleur

Utiliser un autre moyen que la couleur pour transmettre l'information.



Navigation clavier

Permettre la navigation au clavier

Vérifier que le site est utilisable au clavier, que le focus est visible en toute circonstance (propriété **outline** en CSS) et qu'il se déplace de manière séquentiel et logique.

- TAB / MAJ + TAB : pour avancer ou reculer le focus.
- Entrée : pour valider un bouton, un lien...
- Barre espace : pour actionner une case à cocher.
- Flèches directionnelles : pour faire défiler l'affichage, se déplacer dans un menu ou modifier des boutons radio.

Navigation générale

Rendre l'intitulé des liens compréhensible

Les intitulés des liens et boutons doivent être compréhensibles hors contexte. Afin de les rendre plus explicites, on peut compléter avec un attribut `title`.

Masquage accessible

Des éléments peuvent être vocalisés par les lecteurs d'écran sans être affichés. Utiliser l'attribut `aria-label` en reprenant le contenu du `title`, ou utiliser une classe CSS de **masquage accessible** (ex : `visually-hidden` pour le frameworks Bootstrap ou Boosted).

```
<a href="...">valider
  <span class="visually-hidden"> le paiement</span>
</a>
```

Inversement, on peut masquer du contenu pourtant visible à l'écran, celui-ci ne sera pas vocalisé. On utilisera dans ce cas l'attribut `aria-hidden="true"`.

Prévenir en cas d'ouverture de nouvelle fenêtre

Prévenir les utilisateurs des nouvelles fenêtres :

```
<a href="..." title="Consulter les CGU (nouvelle
fenêtre)" aria-label="Consulter les CGU (nouvelle
fenêtre)">Consulter les CGU</a>
```

Fournir des liens d'évitement

Mettre en place des liens d'accès direct permettant d'accéder aux zones principales de la page.

Permettre le contrôle des animations

Tout contenu en mouvement, mis à jour, clignotant, doit pouvoir être stoppé, caché ou mis en pause.

Mise en page

Utiliser des tailles relatives (Responsive Design)

Privilégier des unités relatives (em, rem, %...) permettant l'agrandissement du texte. Rendre fluide la disposition des éléments. Vérifier l'absence de perte d'informations avec un zoom à 200% (disparition ou chevauchement de texte).

Les attributs ARIA

Les attributs et rôles aria permettent d'améliorer la vocalisation de certains éléments pour les personnes qui utilisent un lecteur d'écran. Ils sont sans effet pour les autres utilisateurs.

Les attributs globaux

`aria-label`, `aria-labelledby`

Ajoute une étiquette à l'élément :

```
<button aria-label="rechercher"></button>
<h2 id="search">Rechercher dans le catalogue</h2>...
<button aria-labelledby="search"></button>
```

`aria-describedby`

Ajoute une description à l'élément (vocalisée après le label) :

```
<label for="num">numéro dossier</label>
<input id="num" aria-describedby="desc" type="text">
<span id="desc">3 lettres suivi de 4 chiffres</span>
```

`aria-hidden`

Masque l'élément et ses enfants aux lecteurs d'écran :

```
<button>
  <span aria-hidden="true" class="icon">U</span>
  Rechercher
</button>
```

`aria-haspopup`

Indique la présence d'un menu déroulant :

```
<button aria-haspopup="true">Menu options</button>
```

`aria-expanded`

Indique que l'élément est déplié (menu, section...) :

```
<button aria-expanded="true" aria-
haspopup="true">Menu...
```

`aria-live`

Indique au lecteur d'écran que cet élément doit être vocalisé automatiquement lorsque celui-ci est mis à jour :

```
<span aria-live="polite">Avancement : 50%</span>
```

`disabled` (HTML5)

Indique que l'élément est désactivé :

```
<button disabled>Enregistrer</button>
```

Pour les formulaires

required (HTML5) : indique que l'élément est requis :

```
<input type="text" id="email" required>
```

aria-invalid : Indique une erreur de saisie :

```
<input type="text" id="email" aria-invalid="true">
```

Les rôles

dialog : indique la présence d'une boîte de dialogue :

```
<div role="dialog" aria-labelledby="title">
  <div id="title">...</div>[...]
</div>
```

alert et alertdialog : indique au lecteur d'écran d'annoncer un message important.

```
<p role="alert">Vous devez accepter les CGU</p>
```

La boîte à outils

Les lecteurs d'écran

Quelques lecteurs d'écran pour tester vos sites web :

- Sous Windows, **NVDA** et **JAWS**
- Sous MacOS et IOS, **VoiceOver**
- Sous Android, **TalkBack**

Axe Dev Tools

Extensions disponibles pour Chrome et Firefox permettant d'effectuer des tests automatiques d'accessibilité.

<http://bit.ly/aXe-Firefox> et <http://bit.ly/aXe-Chrome>

Orange Boosted with Bootstrap

Orange Boosted (framework HTML, CSS et Javascript), basé sur Bootstrap. Il permet la mise en place rapide d'un site web responsive et accessible à la charte Orange.

<http://boosted.orange.com>

La Va11ydette

Pour vous aider à vérifier que votre site respecte bien les critères d'accessibilité : <https://la-va11ydette.orange.com>

Accessibilité Numérique Orange

Des recommandations, des outils, des bonnes pratiques proposés par EASE sur l'accessibilité numérique.

<https://a11y-guidelines.orange.com>