

3IN017 – TECHNOLOGIES DU WEB

Accessibilité numérique

25 mars 2025

Gilles Chagnon

Plan

- 1 Le W3C – la WAI**
- 2 L'accessibilité en France**
- 3 Pour qui ?**
- 4 Concrètement...**

Accessibilité « du Web » ?

- matériel

Accessibilité « du Web » ?

- matériel
- logiciel

Accessibilité « du Web » ?

- matériel
- logiciel
- infrastructure réseau

Accessibilité « du Web » ?

- matériel
- logiciel
- infrastructure réseau
- langue maternelle

Accessibilité « du Web » ?

- matériel
- logiciel
- infrastructure réseau
- langue maternelle
- culture

Accessibilité « du Web » ?

- matériel
- logiciel
- infrastructure réseau
- langue maternelle
- culture
- localisation géographique

Accessibilité « du Web » ?

- matériel
- logiciel
- infrastructure réseau
- langue maternelle
- culture
- localisation géographique
- aptitudes physiques

Accessibilité « du Web » ?

- matériel
- logiciel
- infrastructure réseau
- langue maternelle
- culture
- localisation géographique
- aptitudes physiques
- aptitudes mentales

???

Le W3C – la WAI

Le W3C

- Le W3C : organisme chargé de la définition de recommandations pour le Web
- 1996 : création du groupe de travail WAI (*Web Accessibility Initiative*) en charge de la rédaction de guides pour l'accessibilité

Les Guidelines

Types de guidelines (directives) :

- pour les « agents utilisateurs » (User agents), les UAAG ;
- pour les éditeurs, (Authoring Tools), les ATAG ;
- pour les producteurs de contenus (Web contents), les WCAG ;

Nous nous intéresserons surtout aux WCAG.

Les WCAG 1.0

WCAG 1.0, sortie le 5 mai 1999 :

- 14 directives, 65 points de contrôle (*checkpoints*) ;
- 3 niveaux de priorité, 3 niveaux de conformité : A, AA et AAA :
 - A : le « minimum vital ». S'il n'est pas respecté, barrière infranchissable pour certains utilisateurs ;
 - AA : très recommandé. Permet d'améliorer l'accès à certaines informations ;
 - AAA : recommandé. Facilite la consultation des informations.

Les WCAG 2

WCAG 2.0, sortie le 11 décembre 2008, mise à jour 2.2 le 5 octobre 2023

- Généralisent les WCAG 1.0
- Proposent plusieurs grilles de lecture, en fonction des publics :
 - 4 principes
 - 13 règles
 - Des « critères de succès »
 - Des techniques suffisantes et recommandées
 - + liste d'« échecs fréquents documentés »
- En anglais (normatives) : <https://www.w3.org/TR/WCAG22/>

Remarque : Les WCAG 2.3 ont été publiées à l'état de *working draft* le 12 décembre 2024.

Principes

Les contenus doivent être...

perceptibles (*perceivable*)

L'information et l'interface doivent pouvoir être présentées de manière à ce que l'utilisateur puisse les percevoir avec au moins un de ses sens

utilisables (*usable*)

L'interface utilisateur doit pouvoir être... utilisée

compréhensibles (*understandable*)

Les utilisateurs doivent pouvoir comprendre les informations et les éléments de l'interface qui y donnent accès

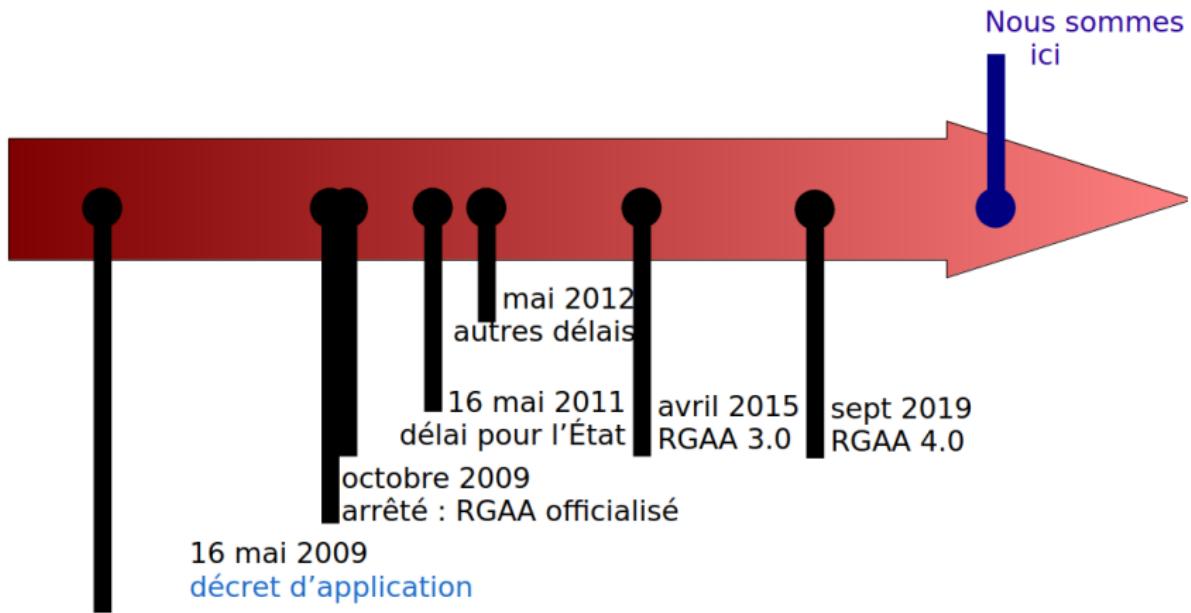
robustes (*robust*)

Le contenu publié doit pouvoir être interprété de manière fiable par une « large variété » d'agents utilisateurs et technologies d'assistance

L'accessibilité en France

Loi « pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées » du 11 février 2005 article 47

- 3 ans à compter de la signature du décret pour la mise en conformité des sites publics
- pas d'obligation pour le secteur privé



Loi du 11 février 2005 :

« Les services de communication publique en ligne des services de l'État, des collectivités territoriales et des établissements publics qui en dépendent doivent être accessibles aux personnes handicapées »



Pour qui ?

Les personnes handicapées

Des ordres de grandeur :

- Handicap moteur 2,3 millions
- Handicap auditif : 4,1 millions (2010, secrétariat d'état chargé de la famille et de la solidarité)
- Handicap cognitif : 700000
- Handicap visuel : 1,7 million (dont 200000 aveugles ou malvoyants profonds) (2005, DREES)
- Multi-handicap : 2,5 millions (2003, ministère de la Santé, de la Famille et des Personnes handicapées)

Source (ancienne !) : https://travail-emploi.gouv.fr/IMG/pdf/donnees_chiffrees_par_handicap.pdf

Handicap visuel : non- et malvoyance



Outils logiciel d agrandissement, loupes, lecteur d écran, synthèse vocale, plage braille

Difficultés contenus graphiques, fonctionnement dépendant de la souris, contrastes de couleur, structure, rechargements intempestifs...

Handicap moteur



Outils reconnaissance vocale, clavier virtuel, dispositifs de pointage spécifiques

Difficultés fonctionnement dépendant de la souris, contenu dynamique, formulaires mal conçus, longueur des pages...

Handicap auditif



Outils vidéo en langue des signes (LSF) ou logiciel de transcription en Langage Parlé Complété (LPC)

Difficultés son, vidéo sans sous-titrage, longues pages. . .

Handicap cognitif

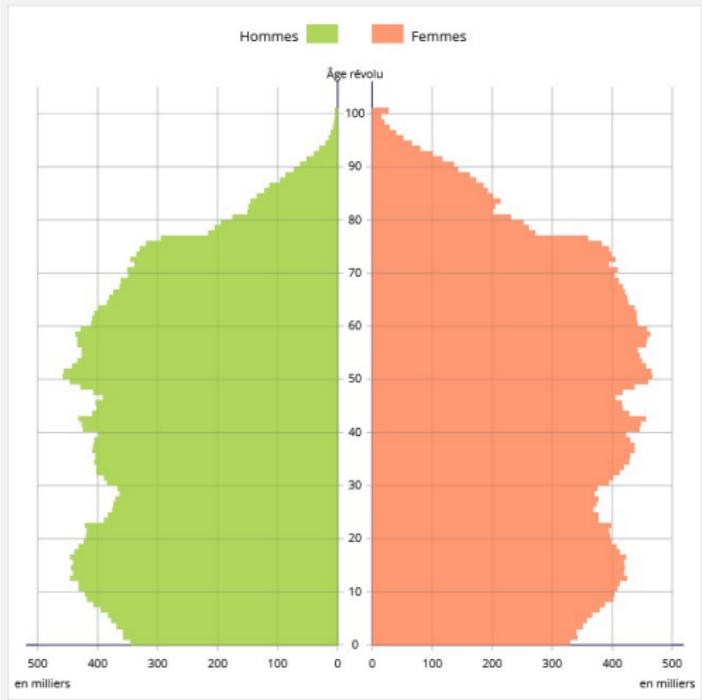


Outils synthèse vocale

Difficultés longues pages, navigation problématique, défilements, animations, clignotements, nouvelles fenêtres, mise en forme du texte... .

... et personnes âgées (France, janvier 2023)

Pyramide des âges en 2023

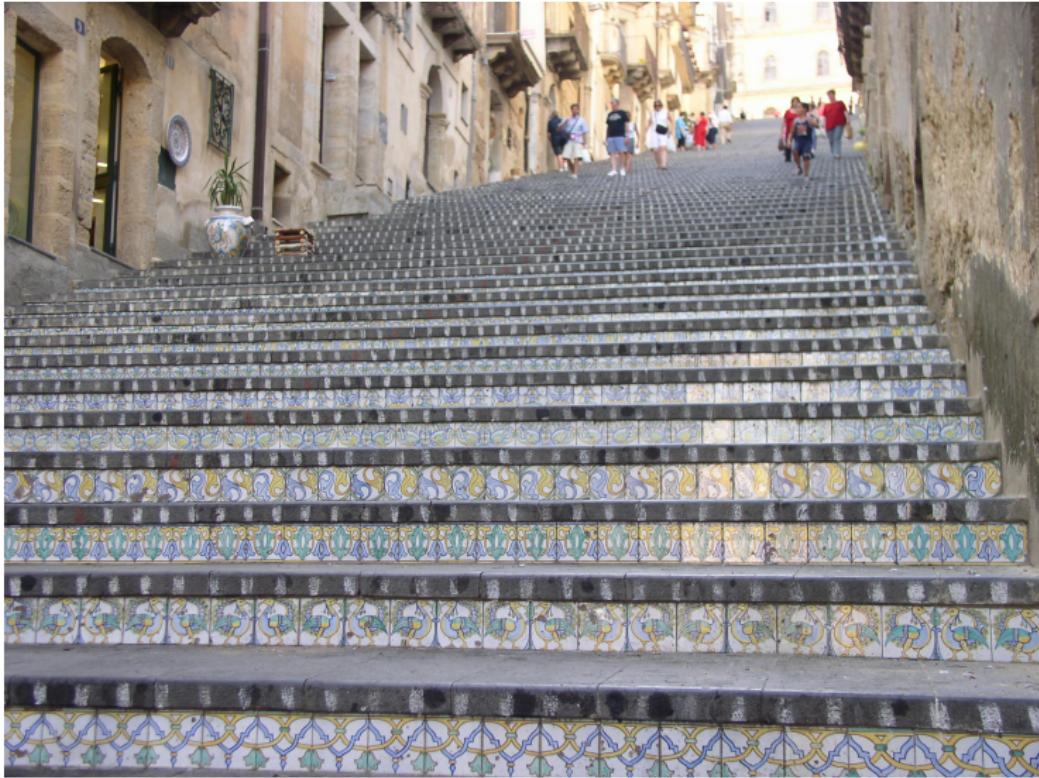


Concrètement . . .

Le RGAA...



Il faut y aller petit à petit



Des « premiers pas » faciles à mettre en œuvre

- Des tests automatisables
- Le minimum nécessaire, qui peut être corrigé sans l'intervention d'un expert
- Prévention des erreurs
- https://res.cloudinary.com/opquast/image/upload/v1556789103/checklists/Opquast-steps_20120330.pdf

Code conforme

- Utiliser une DTD conforme aux recommandations du W3C
- Ne pas utiliser de syntaxe d'attribut erronée ni d'imbrication d'élément non conforme

Ne pas faire !

```
<img source="ours.png" alt = "ours" />
<ul>
    <li></li>
    <ul>
        <li></li>
    </ul>
</ul>
```

- Pas d'élément obsolète : ~~xmp, listing, tt, plaintext, strike, basefont, center, font~~
- Pas d'attribut obsolète : ~~align, alink, vlink, link_text, border, background, bgsound, bgcolor~~
- Pour les listes : au moins un élément `dt` enfant de `dl`, pas de `li` vide...
- Pas de paragraphe vide, pas de lien à vide (sauf les ancrés ``), pas de `button` vide

Pas d'ambiguïté

Liens

- Les intitulés de lien doivent être explicites en contexte (pas plusieurs voir le message dans une même page ; mais on peut préciser un attribut title : voir le message)
- Pas de lien avec le même intitulé mais pointant vers des ressources différentes

Acronymes et abréviations doivent être explicités (attribut title ou en plein texte)

Les titres

Titre de la page utiliser l'élément `title` dans le `head`, qui doit être pertinent

Hiérarchie de titres ■ Au moins un `h1` dans `body`

- Pas de titre `h1`... `h6` vide
- Respecter la hiérarchie des titres (par exemple, pas de `h4` après un `h2`)

Changements de langue

- Utiliser l'attribut `lang` avec une valeur normalisée
- Utiliser `dir="rtl"`, `dir="ltr"` ou `dir=""` et rien d'autre si nécessaire

Alternatives textuelles

- Utiliser alt pour img, area, applet, input type="image"
- alt non vide et pertinent pour :
 - img seul enfant de a ou button
 - input type="image"
 - area avec attribut href
- Pas d'information introduite dans la CSS

Couleurs

- Pas d'information portée uniquement par la couleur
- Attention aux contrastes

Tableaux de données

- Attribut scope ou id sur chaque th non vide
- scope="row" ou scope="col"

Animations et vidéos

- en CSS
- `marquee`, `blink`
- vidéos : sous-titrées, audio-description, transcription textuelle

Formulaires

- Utiliser un bouton ou un `input type="submit", "image" ou "button"` dans chaque form
- Pas d'attribut `value` vide sur un submit :
`<input type="submit" value="" />`
- `fieldset` uniquement dans les forms
- Utiliser `legend`, non vide, quand il y a un `fieldset`
- Associer systématiquement un élément `label` à chaque champ de formulaire à remplir (éléments `input`, `select`, `textarea`...)

```
<label for="age">Âge</label>
<input type="number" id="age" min="0">
```

Clavier et souris

- Toute action réalisable avec la souris doit aussi être réalisable avec le clavier

Pour *Accessible Rich Internet Applications* car les documents HTML sont devenus plus complexes. Trois composants :

- des rôles pour spécifier ou parfois dupliquer le rôle joué par un élément (par exemple `<nav role="navigation">`)
- des propriétés pour ajouter une propriété supplémentaire ou de la sémantique à un élément. Par exemple, `<input id="champ_code" aria-required="true">` pour préciser qu'un champ de saisie est obligatoire.
- des états pour définir l'état actuel d'un élément (par exemple `aria-disabled="true"` si un champ de formulaire est désactivé).

Ce sont des surcouches pour compléter l'accessibilité du code HTML lui-même, et qui reposent sur le support de JavaScript par les outils de consultation.

