

Mémoire de recherche



Quid de l'accessibilité au numérique des personnes en situation de handicap ?

Présenté par : Adrien PIECHOCKI

Licence Pro Web et Applications Mobiles

Sous la direction de : François-Xavier MICHAUD et Jean-Luc BOURDON



Remerciements

Je tiens à remercier pour leurs enseignements et leur soutien mes professeurs, Messieurs Jean-Luc Bourdon et François-Xavier Michaud, qui m'ont transmis leurs savoirs durant ces années de Licence pro Web et Applications Mobiles et qui, par leurs initiatives en faveur des élèves en situation de handicap, participent, avec le concours de M. Parent, référent handicap de l'université de Cergy, à rendre l'université plus inclusive.

Je remercie aussi Mme Courtois, pour son suivi des relations entre l'université de Cergy et les entreprises, qui m'a transmis une offre de stage proposée par la MAS Perce-Neige de Boulogne-Billancourt à laquelle j'ai candidaté. Ce fut le début d'un partenariat enrichissant qui s'est concrétisé par un contrat d'apprentissage, aujourd'hui renouvelé.

Un grand merci à Mesdames Valérie de Saint-Laurent et Nathalie Estin-Boric, directrices de la MAS Perce-Neige de Boulogne-Billancourt, qui m'ont, tour à tour, fait confiance pour proposer des services numériques efficaces aux résidents en Locked-In Syndrome.

Je tiens également à remercier Mme Héloïse Ducept et M. Guilhem Bourdeau, mes maîtres de stage et d'apprentissage, qui m'ont guidé auprès des résidents de la MAS Perce-Neige et initié à la Communication Alternative Améliorée et à ses logiciels.

J'exprime ma reconnaissance envers M. Antoine LeGoff, ergothérapeute, pour son accompagnement. Nous avons travaillé ensemble quotidiennement pour prendre en charge les besoins des résidents.

Je tiens aussi à adresser mon soutien et ma gratitude aux résidents de la MAS Perce-Neige de Boulogne-Billancourt qui m'ont accueilli et sollicité pour rendre leurs outils numériques plus opérationnels et m'ont ainsi confronté aux limites des technologies d'assistance.

Enfin, merci à mes parents, Laurent et Delphine Piechocki pour leur soutien constant et leurs encouragements.

Sommaire

Introduction	3
I) Qu'est-ce que l'accessibilité numérique ? : définition, personnes concernées et enjeux	4
1) Que signifie « accessibilité numérique » ?	4
2) Quelles sont les personnes principalement concernées et à quelles difficultés sont-elles confrontées ?	4
3) Quels sont les enjeux de l'accessibilité numérique pour les organisations et entreprises ?	6
II) Que dit la loi ? : normes et référentiels	7
1) Les recommandations internationales	7
2) Les lois françaises	8
III) Quels sont les outils technologiques d'assistance existants ? : panorama et limites	9
1) Pour les personnes ayant des déficiences visuelles	9
2) Pour les personnes ayant des troubles de l'audition	10
3) Pour les personnes ayant des troubles de la parole	10
4) Pour les personnes paralysées ou à mobilité réduite	10
5) Pour les personnes ayant des troubles cognitifs	10
IV) Comment faciliter l'accès des personnes en situation de handicap à un site web ?	12
1) Créer un site web accessible : recommandations	12
a) En matière de structuration	12
b) En matière de design	17
c) En matière d'interactivité	22
2) Tester et rendre un site existant plus accessible	24
a) Méthodologie de test : les principaux points à vérifier	24
b) Outils automatisés de test existants	28
c) Expérience utilisateur et expertise	32
d) Corrections et déclaration de l'accessibilité	33
V) Quid de l'accessibilité des applications mobiles ?	34
1) L'accessibilité sur IOS et Androïd	34
2) Le responsive design	35
3) Des exemples réussis d'aide à l'autonomie	37
Conclusion	38
Bibliographie et sitographie	39

Introduction

Depuis la démocratisation du numérique, les technologies, qu'elles soient matérielles (téléviseurs, smartphones, tablettes, ordinateurs) ou logicielles (sites internet, applications mobiles) se sont multipliées.

Étudiant en situation de handicap, en difficulté avec l'écriture manuscrite, j'ai pu trouver, grâce à ces outils, des solutions qui m'ont permis de poursuivre mes études jusqu'en Licence pro Web et Applications Mobiles. Celle-ci se conclut avec la rédaction de ce mémoire.

Ma maladie me rend aussi particulièrement attentif à la problématique du handicap et aux personnes qui y sont confrontées, que ce soit temporairement ou durablement. C'est pourquoi je me suis posé la question suivante : qu'en est-il de l'accessibilité au numérique des personnes en situation de handicap ? Les ressources en ligne permettent de se former, se cultiver, effectuer des démarches administratives, mais sont-elles toutes utilisables par les 12 millions de Français en situation de handicap ? Et que faut-il faire pour qu'elles le soient ?

J'ai consulté de nombreux sites internet, livres et documents officiels pour répondre aux interrogations qui constituent le corps de ce mémoire : que signifie « accessibilité numérique » ? Quelles sont les personnes concernées et à quelles difficultés sont-elles confrontées ? Quels sont les enjeux de l'accessibilité numérique pour les organisations et entreprises ? Existe-t-il des lois à ce sujet et qu'imposent-elles ? Quels sont les outils technologiques d'assistance existants ?

En apprentissage, dans une Maison d'Accueil Spécialisée qui propose ses services à des résidents lourdement handicapés, pour la plupart atteints du Locked-In Syndrome, j'ai pu tester certains outils, en mesurer l'impact et les limites, me confronter à l'inaccessibilité de nombreux sites internet. C'est pourquoi, en lien direct avec ma formation et cette expérience, j'ai choisi de centrer mes recherches sur l'accessibilité du Web. J'ai donc rassemblé dans ce mémoire des recommandations pour créer un site web accessible ou pour tester et transformer un site existant de manière à le rendre plus accessible. Enfin, je me suis penché sur les applications mobiles et j'ai fourni quelques exemples réussis d'aide à l'autonomie, en espérant que beaucoup d'autres viennent, à l'avenir, faciliter la vie des personnes en situation de handicap.

I) Qu'est-ce que l'accessibilité numérique ? : définition, personnes concernées et enjeux

1) Que signifie « accessibilité numérique » ?

L'accessibilité numérique est un droit fondamental : c'est la possibilité pour tous d'accéder aux outils numériques, quelles que soient les limitations physiques, technologiques ou environnementales.

L'accessibilité web désigne l'ensemble des techniques permettant de rendre un site internet accessible à tous, et notamment aux personnes en situation de handicap.

La WAI (Web Accessibility Initiative), instance officielle en charge de cette question, créée en 1997 par le W3C (World Wide Web Consortium), donne cette définition : « L'accessibilité du Web signifie que les personnes en situation de handicap peuvent utiliser le Web. Plus précisément, qu'elles peuvent percevoir, comprendre, naviguer et interagir avec le Web, et qu'elles peuvent contribuer sur le Web. Cette accessibilité concerne tous les handicaps qui affectent l'accès au Web, ce qui inclut les handicaps visuels, auditifs, physiques, d'expression, cognitifs et neurologiques. L'accessibilité du Web bénéficie aussi à d'autres, notamment les personnes âgées dont les capacités changent avec l'âge. »

La notion de handicap s'est en effet progressivement élargie à toutes les formes de contraintes, durables ou temporaires. Le but est de répondre aux besoins et préférences de chaque utilisateur.

Au-delà du web, ce principe s'applique à tous les types d'interfaces numériques : applications mobiles, logiciels, télévision, domotique, jeux vidéo, bornes interactives, etc.

En résumé, les services et informations numériques doivent être accessibles et utilisables par tous, quels que soient leur handicap, le contexte d'usage (luminosité inadaptée à l'écran, environnement imposant le silence, faible connexion internet) ou le matériel utilisé (PC, tablette, téléphone mobile, montre connectée).

2) Quelles sont les personnes principalement concernées et à quelles difficultés sont-elles confrontées ?

L'Organisation Mondiale de la Santé estime que 1,3 milliard de personnes, soit 1 personne sur 6 dans le monde, sont atteintes d'un handicap important. Dans l'Union européenne, environ 87 millions de personnes et en France, 12 millions, vivent avec un handicap plus ou moins sévère. Cela correspond à près de 20% de la population et peut augmenter jusqu'à 40% si l'on tient compte des situations de handicap temporaires.



- a) Les personnes ayant un handicap visuel : aveugles, malvoyantes, ayant des difficultés à distinguer les couleurs (daltonisme) ou sensibles à une trop forte luminosité, peuvent avoir du mal à déchiffrer les textes trop petits ou pas assez contrastés (ex : gris sur fond noir). Les liens hypertextes et boutons non explicités sont aussi souvent difficiles à percevoir.
- b) Les personnes ayant un handicap auditif : sourdes, malentendantes, ayant des difficultés de perception de la parole dans un environnement bruyant, ne peuvent accéder aux contenus audio et vidéo sans sous-titres ou interprète en langue des signes. Elles n'entendent pas les messages ou signaux uniquement sonores. La surdité profonde entraîne une impossibilité à acquérir le langage oral et complique l'apprentissage de la lecture. De ce fait, le sous-titrage d'un élément audio n'est pas toujours suffisant et il faut penser à proposer une version doublée en langue des signes.
- c) Les personnes ayant un handicap de parole : muettes, bégues ou ayant des troubles de l'articulation, ne peuvent utiliser la reconnaissance vocale.
- d) Les personnes ayant un handicap moteur. Avec une aptitude limitée ou une impossibilité à mouvoir certaines parties du corps, à se déplacer, à contrôler leurs mouvements (tremblements) ou à les effectuer de façon précise, ces personnes peuvent avoir des difficultés à utiliser une souris, un clavier ou un écran tactile, ne pas pouvoir cliquer, réaliser des combinaisons de touches, enchaîner des actions séquentielles, et donc naviguer facilement sur un site.

- e) Les personnes ayant un handicap cognitif : dyslexiques, personnes souffrant de troubles du spectre autistique ou d'autres déficiences affectant la lecture, la mémorisation, la concentration, la compréhension, l'analyse des situations, peuvent avoir des difficultés à se repérer et à naviguer dans un site qui n'est pas intuitif (intitulés ou icônes peu clairs, blocs de texte trop longs et mal structurés, arborescence complexe, fenêtres qui se superposent). Elles ne parviennent alors pas à trouver l'information recherchée. L'usage des périphériques numériques (téléphone, ordinateur, tablette, clavier, souris) peut être difficile, ainsi que le remplissage de formulaires.

Ces handicaps peuvent se cumuler : sourds-muets, sourds-aveugles, personnes polyhandicapées suite à un AVC, atteintes de Locked-In-Syndrome, etc.

Cela concerne aussi les personnes les moins à l'aise avec la technologie et notamment les personnes âgées qui, en vieillissant, peuvent avoir une baisse de leurs facultés visuelles, auditives et motrices.

Enfin, l'accessibilité numérique est un problème qui peut toucher tout le monde à un moment de sa vie (accident, opération, blessure, immobilisation créant un handicap temporaire ou permanent).

3) Quels sont les enjeux de l'accessibilité numérique pour les organisations et entreprises ?

Environ 12 millions de Français sont concernés par le handicap. Les prendre en compte permet :

- d'œuvrer pour l'inclusion en garantissant à tous un accès égal aux outils numériques,
- de toucher un public plus large et donc d'augmenter le trafic sur son site, avec parfois des retombées financières, les personnes souffrant de handicap ayant l'habitude de faire leurs achats sur internet,
- d'optimiser le référencement naturel (SEO),
- d'améliorer l'expérience utilisateur, handicapé ou non.

C'est donc l'image de l'entreprise et son attractivité qui sont renforcées. D'autant que les améliorations réalisées peuvent bénéficier à tous. Ainsi, l'accessibilité peut apporter du confort aux personnes non handicapées dans les situations suivantes :

- les environnements bruyants tels que les open spaces, boutiques, halls d'accueil...
- des conditions de connexion dégradées (internet bas débit)
- des formulaires à renseigner : la navigation au clavier permet de remplir les formulaires sans utiliser la souris et apporte un gain de temps appréciable.

L'accessibilité est donc un enjeu de citoyenneté. C'est aussi désormais une obligation légale.

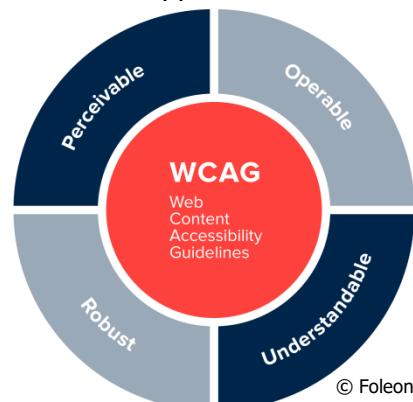
II) Que dit la loi ? : normes et référentiels

1) Les recommandations internationales

Au niveau international, le Web Content Accessibility Guideline, ou WCAG 2.1, fait référence. Ses directives, qui résultent d'un consensus à l'échelle mondiale entre les spécialistes du World Wide Web Consortium (W3C), évoluent depuis plus de 20 ans et s'appliquent à tout type de contenus diffusables via le Web. Ce guide de bonnes pratiques constitue un outil primordial pour la mise en accessibilité des sites web et des applications mobiles.

Le WCAG s'articule autour de 4 piliers :

- la perception (perceivable),
- l'utilisabilité (operable),
- la compréhension (understandable),
- la compatibilité (robust).



Voici les 12 principales recommandations :

- Perception #1 : proposer des équivalents textuels à tout contenu non textuel.
- Perception #2 : proposer des sous-titres ou des versions de remplacement pour tous les contenus multimédias.
- Perception #3 : rendre les contenus adaptables. Par exemple, proposer une version audio des contenus, donner la possibilité de modifier la taille de la police...
- Perception #4 : jouer sur les contrastes pour permettre de bien distinguer les différents contenus (visuels, audio...) du site.
- Utilisabilité #1 : faire en sorte de rendre les fonctionnalités accessibles au clavier (raccourcis).
- Utilisabilité #2 : laisser le temps à l'internaute de lire le contenu qui lui est présenté.
- Utilisabilité #3 : ne pas utiliser de contenu susceptible de provoquer des crises convulsives.
- Utilisabilité #4 : aider l'utilisateur à naviguer dans les pages et contenus et à se localiser sur le site (moteur de recherche, menu, fil d'Ariane).
- Compréhension #1 : rendre les contenus textuels lisibles et compréhensibles.
- Compréhension #2 : faire en sorte que les contenus apparaissent et fonctionnent de manière prévisible.
- Compréhension #3 : aider l'internaute à éviter et à corriger les erreurs.
- Compatibilité #1 : optimiser le site afin qu'il soit compatible avec les technologies actuelles et futures.

La norme WCAG définit trois niveaux d'accessibilité :

- A : accessibilité correcte
- AA : accessibilité optimisée
- AAA : accessibilité excellente

Sur la base de ce WCAG, certains pays, comme les États-Unis, l'Allemagne ou la France ont mis en place leurs propres référentiels.

2) Les lois françaises

En France, plusieurs textes sont à prendre en compte. L'accessibilité y est une obligation légale :

- 2005 : loi du 11 février dite « loi handicap » : cette loi concernant « l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées », impose à l'ensemble des administrations la mise en accessibilité de leurs sites web. Elle est explicite : « Les services de communication publics en ligne des services de l'État, des collectivités territoriales et des établissements publics qui en dépendent doivent être accessibles aux personnes handicapées. L'accessibilité des services de communication publics en ligne concerne l'accès à tous types d'information sous forme numérique, quels que soient le moyen d'accès, les contenus et le mode de consultation. Les recommandations internationales pour l'accessibilité de l'internet doivent être appliquées pour les services de communication publique en ligne. » Le secteur privé est aussi concerné au sens où les employeurs doivent veiller à une égalité de traitement concernant l'emploi des travailleurs handicapés (accès, formation...).
- 2006 : La convention des Nations Unies relatives aux droits des personnes handicapées (CDPH) reconnaît un accès égal aux technologies de la communication.
- 2009 : Publication de la première version du RGAA : Référentiel Général d'Accessibilité pour les Administrations. Ce RGAA 1.0 s'inspire des WCAG. Le volet juridique contient une liste de sanctions.
- 2010 : L'Union européenne lance la stratégie 2010-2020 en faveur des personnes handicapées (European Disability Strategy).
- 2016 : Loi pour une République numérique : obligation pour les administrations d'établir un schéma pluriannuel de mise en accessibilité et d'afficher, sur la page d'accueil de leur site, leur conformité aux règles d'accessibilité. Trois apparaissent prioritaires : la circulation des données et du savoir, la protection des individus dans la société du numérique, l'accès au numérique pour tous.
- 2019 : Parution de la 4e version du RGAA, prenant en compte les évolutions de la norme WCAG. Le Référentiel Général d'Accessibilité pour les Administrations (RGAA) devient le Référentiel Général d'Amélioration de l'Accessibilité. Cette nouvelle version concerne désormais des entreprises privées, et plus seulement les administrations. Suivant les obligations légales du décret de juillet 2019, la loi étend l'obligation d'accessibilité web aux personnes morales de droit privé ayant une mission de service public ou d'intérêt général et à toutes les entreprises faisant un chiffre d'affaires annuel de plus de 250 millions d'euros. Celles-ci ont l'obligation de publier une déclaration de conformité. Le défaut d'affichage de la conformité de leur site est passible de 20 000 euros d'amende par an et par application. Cette 4e version du RGAA décrit les responsabilités des différents acteurs en matière d'accessibilité (experts web, responsables de la politique du handicap, chefs d'entreprise) et répertorie les normes d'accessibilité du RGAA et les tests permettant de vérifier la conformité d'une page web. L'année 2019 voit aussi la mise en place de l'observatoire de la qualité des services numériques, qui suit la qualité des démarches et services numériques de l'Etat, notamment en termes de prise en compte du handicap.

III) Quels sont les outils technologiques d'assistance existants ? : panorama et limites

Certaines personnes ont besoin de technologies d'assistance (matérielles et logicielles) pour accéder au monde numérique et interagir avec son contenu. Nombreux mais parfois méconnus, ces outils connaissent de constantes évolutions.

1) Pour les personnes ayant des déficiences visuelles, il est possible

d'accéder aux contenus du Web grâce à la technologie de la synthèse vocale (logiciel qui lit le contenu de la page à voix haute).

Pour les aveugles, des lecteurs d'écran peuvent retranscrire l'information vocalement ou en braille. Les plus courants sont [NVDA](#) (opensource et optimisé pour Firefox), [JAWS](#) (payant et optimisé pour Internet Explorer), [VoiceOver](#) (intégré à MacOS et iOS), [Talkback](#) pour Androïd, [ChromeVox](#) (Chrome) et [Orca](#) (Linux).

Lorsque le lecteur d'écran est associé à une plage braille, les informations transmises s'affichent sous forme de points en relief, décryptables par la personne aveugle.

À noter : les logiciels de lecture d'écran peuvent également être utilisés par des personnes voyantes mais qui sont en difficultés avec la lecture (dyslexiques, dyspraxiques).

Pour les malvoyants, parfois des personnes âgées victimes de cataracte ou de DMLA (dégénérescence maculaire liée à l'âge), et peu à l'aise avec les outils informatiques, des claviers gros caractères, des loupes d'écran ou des logiciels permettent d agrandir les textes ou de zoomer sur une partie de l'écran. Certains retrouvent aussi vocalement l'information. Ils ont également d'autres options utiles qui facilitent la lecture et diminuent la fatigue oculaire : amélioration du pointeur de la souris ou du curseur texte, modification des couleurs et contrastes en fonction des besoins.

Les logiciels d'agrandissement les plus répandus sont [SuperNova](#) et [ZoomText](#) qui peut zoomer jusqu'à 36 fois. Pour les déficiences visuelles légères, les zooms des navigateurs ou celui de Windows peuvent être suffisants.

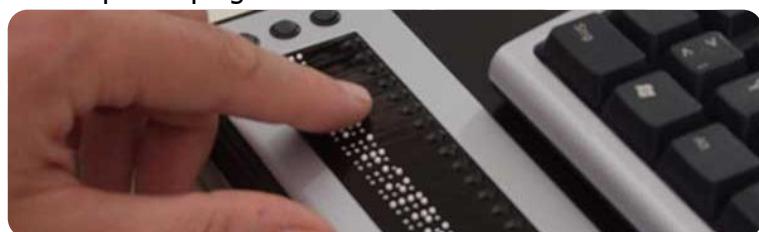
La vocalisation (ex : [ReadSpeaker](#)) est également très utile.

Tous ces dispositifs sont performants, mais leur confort d'utilisation dépend vraiment des efforts de conception déployés par les développeurs des sites web sur lesquels ils sont utilisés. Si leur usage n'est pas pris en compte lors du développement, ces outils peuvent s'avérer totalement inutiles.

Enfin, les livres numériques au format audio ou en grands caractères permettent de maintenir l'accès à la lecture.

Récente innovation, Microsoft propose [Canetroller](#), un outil rendant accessible la réalité virtuelle aux malvoyants grâce à la simulation haptique (faisant ressentir des vibrations légères) et auditive d'une canne.

Exemple de plage braille :



© Ceciaa

- 2) Pour les personnes ayant des troubles de l'audition**, les audioprothèses de dernière génération, connectables en Bluetooth, permettent de transmettre le son depuis un ordinateur ou un smartphone. Pour les personnes sourdes, des traductions sont possibles en langue des signes par visioconférence. Des solutions de reconnaissance vocale ont été adaptées pour retranscrire le langage parlé en langage écrit (ex : [AVA](#), [RogerVoice](#)). Une transcription textuelle des contenus multimédias est souvent indispensable (ex : [AUTHOT](#) : retranscription audio automatique et sous-titrage vidéo). Incrustations et pictogrammes améliorent aussi la navigation.
- 3) Pour les personnes ayant des troubles de la parole**, des outils de communication alternative améliorée (CAA) ont été développés (Ex : [Grid](#), [Communicator](#)). Ils ont pour fonction primaire d'oraliser un texte tapé au clavier.
- 4) Pour les personnes paralysées ou à mobilité réduite**, il existe des claviers spécifiques, des souris ergonomiques ou gyroscopiques, des trackpads alternatifs, des joysticks, des guide-doigts, des contacteurs qui se fixent sur une partie du corps ayant conservé une capacité de mouvement, des outils de suivi de l'œil (eye-tracking) avec caméra (cf. [Cenomy](#)) ou des capteurs spécifiques avec possibilité de clic par mouvement des paupières ([Wyes](#)), des logiciels de reconnaissance faciale, des assistants vocaux capables de comprendre la voix des utilisateurs et de répondre efficacement à leurs requêtes. Pour être facilitée, la navigation doit pouvoir se faire uniquement au clavier (réel ou à l'écran). Certains paramétrages (ex : réglage de la vitesse de la souris) peuvent aussi améliorer l'accessibilité.

Exemple de lunettes connectées :



© Wyes

- 5) Pour les personnes ayant des troubles cognitifs**, il existe des logiciels de dictée et d'écriture avec prédition de mots (ex : [Lexibar](#)), des didacticiels utilisant des icônes (ex : [Auticiel](#)), des applications de gestion d'emploi du temps (ex : [Watchhelp](#)). Les supports tactiles (tablette) sont souvent plus maniables. La méthode FALC (Facile à Lire et à Comprendre) et les polices « sans serif » sont à privilégier. La tolérance à l'erreur pour les formulaires à remplir peut être augmentée avec des messages d'aide suggérant des solutions.

Globalement, le numérique et ses nombreuses applications (santé, domotique, loisirs) améliorent le confort de vie des personnes en situation de handicap et leur autonomie. Il permet de faire des démarches administratives, des consultations médicales, des formations ou des rencontres à distance, de contrôler certaines fonctions de leur logement (chauffage, fermeture des volets), de télétravailler, de visionner des films. Face aux problèmes de mémoire, téléphones et montres connectées permettent de lancer des rappels et des alertes. Cependant, toutes ces technologies ont un coût qui peut être rédhibitoire, même si certaines MDPH (Maison Départementale des Personnes Handicapées) et associations peuvent en financer une partie. Enfin, ces outils d'assistance peuvent vite devenir inefficaces lorsqu'un site web ou une application ne respecte pas les bonnes pratiques d'accessibilité numérique.

Deux exemples concrets de limites au fonctionnement de ces outils :

- le dysfonctionnement du lecteur d'écran :

Un logiciel de lecture d'écran parcourt tout le contenu textuel d'un site. L'utilisateur le contrôle au clavier pour se déplacer sur la page, il utilise les liens et menus de navigation et il lit la description des images. Mais difficile de s'y retrouver quand on ne comprend pas comment la page est construite et à quoi correspond chaque élément. Les titres peu clairs, les onglets, les liens qui ouvrent une nouvelle fenêtre sans possibilité de retour en arrière, les effets d'animation (carrousels, listes déroulantes) peuvent poser problème. Enfin, la lecture des images peut juste se résumer à « img1.jpg ».

- les problèmes avec les logiciels d agrandissement :

Les malvoyants qui les utilisent agrandissent aussi chaque petit défaut. Les problèmes de contraste sont très gênants. Un texte peu contrasté ne se détache pas assez du fond et est difficile à déchiffrer. Enfin, en zoomant fortement, les logiciels d agrandissement bousculent la mise en page que le concepteur a soigneusement calibrée, jusqu'à la rendre complètement illisible.

Exemple de textes superposés :

basse saison :	480 EUR - Prix pour 3 nuits par chambre
basse saison :	965 EUR - prix par semaine par chambre
moyenne saison :	14242EURprirprparsemairpeahabambre
haute saison :	9696EURprirprparsemairpeahabambre
haute saison :	19193EURprirprparsemairpeahabambre
très haute saison :	25863EURprirprparsemairpeahabambre
*Les prix sont indiqués à titre indicatif.	
<u>basse saison:</u>	
du 18-01 au 14-02-2012	
du 01-02 au 20-02-2012	
<u>haute saison:</u>	
du 04-01 au 07-02	
du 08-03 au 19-04	
<u>moyenne saison:</u>	

© OpenClassrooms

À noter : les PDF ne sont pas toujours utilisables avec les logiciels d agrandissement, car les lecteurs de PDF ne permettent pas automatiquement de réajuster la taille des lignes au contenu.

IV) Comment faciliter l'accès des personnes en situation de handicap à un site web ?

1)Créer un site web accessible : recommandations

On doit avoir en tête quand on développe un nouveau site web que certaines personnes ont besoin d'utiliser un des outils précédemment cités et notamment, de manière assez fréquente, un lecteur d'écran. Deuxième point important : nombreux sont ceux qui, pour des raisons diverses, ont des difficultés à utiliser la souris. En termes de développement web, deux objectifs de base doivent donc être visés :

- L'information doit être accessible aux logiciels de lecture d'écran.
Si le site n'est pas accessible : le logiciel ne lira pas correctement. Il ne permettra donc pas à son utilisateur de trouver l'information qu'il cherchait.
- La navigation doit pouvoir se faire sans la souris, avec juste une touche du clavier (ou un contacteur). Il faut pouvoir dérouler le texte de ligne en ligne seulement avec la flèche descendante et naviguer entre les différents champs d'un formulaire avec la touche Tab. Globalement, toutes les fonctions du site doivent être accessibles au clavier. Cependant, comme certaines personnes peuvent aussi avoir des difficultés à bien utiliser le clavier, il faut prévoir une tolérance à l'erreur.

Globalement, voici les principales recommandations à suivre pour créer un site web dans une optique d'accessibilité :

a) En matière de structuration

La logique du code source est primordiale, elle doit être en adéquation avec la structuration.

- **La structuration du site :**

Il faut veiller à ce que la profondeur du site, c'est-à-dire le nombre de clics nécessaires pour accéder à la page voulue depuis la page d'accueil, soit la plus faible possible. En effet, une personne rencontrant des difficultés doit pouvoir accéder rapidement et simplement à l'information souhaitée. La structuration est donc primordiale, notamment pour les personnes utilisant un lecteur d'écran. Au-delà de la lecture du contenu de la page, celui-ci permet de naviguer d'un titre ou d'un lien à l'autre. Cette navigation n'est possible que si le site et les pages sont structurés de façon logique. L'ordre du code doit donc refléter l'ordre de lecture.

Les balises ont un rôle important. Le langage HTML utilise des balises sémantiques, c'est-à-dire ayant du sens. Elles permettent de structurer les informations sur le site. Il faut donc utiliser des éléments de sectionnement appropriés pour envelopper la navigation principale (`<nav>`). Les balises fournissent une sémantique supplémentaire aux lecteurs d'écran et donnent à l'utilisateur des indices sur le contenu au sein duquel il navigue, ce qui lui permet de trouver ce qu'il recherche plus rapidement.

Pour rendre ce contenu encore plus accessible, on peut placer sur les balises HTML des attributs ARIA (Accessible Rich Internet Applications). Ils permettent aux lecteurs d'écran de savoir comment interagir avec les balises en question quand les fonctionnalités standards ne le permettent pas.

- Voici des exemples des attributs ARIA les plus utiles :
 - aria-label : il permet d'indiquer au lecteur d'écran quel texte doit être lu au passage sur l'élément HTML auquel il est rattaché.

Exemple :

```
<button type="button" aria-label="Ce texte sera lu par les lecteurs d'écran">Bouton avec aria-label</button>
```

 - aria-live : cet attribut prévient le lecteur d'écran d'un changement, comme l'apparition d'un pop-up. Il y a 3 valeurs pour définir le niveau de priorité du texte à lire :
 - off : c'est la valeur par défaut. Le texte sera ignoré et l'utilisateur ne sera pas notifié.
 - polite : le lecteur d'écran attendra d'avoir terminé une lecture en cours avant de lire le contenu du nouvel élément.
 - assertive : le lecteur d'écran lira cet élément en priorité. S'il est en train de lire autre chose, il s'interrompt pour lire celui-ci, ce qui est utile dans le cas d'un pop-up d'alerte par exemple.
 - aria-atomic : il indique que le contenu de l'élément doit être lu comme une seule entité. Sans cela, seule la partie du contenu qui change sera lue.
Par exemple, si on passe de 2 à 3 utilisateurs, avec ce bloc HTML :

```
<div id="nb-user-info" aria-live="polite">
  {{nbUser}} utilisateurs correspondants
</div>
```

Sans le aria-atomic, le texte qui sera lu au changement du nombre d'utilisateurs sera "3". Si on modifie ce bloc pour y ajouter un aria-atomic="true" :

```
<div id="nb-user-info" aria-live="polite" aria-atomic="true">
  {{nbUser}} utilisateurs correspondants
</div>
```

le texte lu sera alors "3 utilisateurs correspondants".

 - role : le nom de ces rôles ARIA donne des indications aux lecteurs d'écrans sur le but de l'élément. Voici quelques exemples :
 - alert : indique qu'un élément apparaît à l'écran dynamiquement, comme un pop-up. Cela permet aux lecteurs d'écran de le lire en priorité.
 - button : peut être utilisé pour indiquer aux lecteurs d'écran qu'il s'agit d'un élément cliquable.
 - checkbox : indique aux lecteurs d'écran qu'il s'agit d'un élément que l'on peut cocher. Il faut l'utiliser avec aria-checked pour indiquer l'état de cet élément.
 - dialog : indique que l'élément est une fenêtre qui s'ouvre par-dessus l'écran.

Les messages de statut qui informent du résultat d'une action ou de l'existence d'erreurs doivent pouvoir être correctement restitués par les technologies d'assistance. Pour cela, il faut utiliser les attributs ARIA `role = "status"` dans le premier cas et `role = "alert"` en cas d'erreur.

- L'attribut tabindex :

Par défaut, tous les éléments avec lesquels un utilisateur peut interagir sont accessibles via la touche de tabulation. Ces éléments sont les boutons, les inputs et les liens. Il peut arriver que l'on ait besoin de rendre un élément atteignable avec la tabulation, ou simplement focalisable. Pour cela, on peut utiliser l'attribut `tabindex`. Cet attribut permet d'indiquer aux navigateurs que l'élément auquel il est rattaché est focalisable. Sa valeur est un nombre et indique l'ordre dans lequel il doit être atteint. Voici un exemple :

```
<div tabindex="1">Texte 1</div>
<div tabindex="3">Texte 3</div>
<div tabindex="2">Texte 2</div>
```

Sans les `tabindex`, aucun de ces éléments ne pourrait être atteint avec la tabulation, car la balise `<div>` n'est pas focalisable par défaut. Avec les `tabindex`, lors d'un déplacement avec la tabulation, l'utilisateur se déplacera dans l'ordre indiqué par la valeur de chacun d'eux. On aura donc Texte 1, puis Texte 2 et enfin Texte 3. Cependant, si l'on vient ajouter un bouton (qui est nativement focalisable), le navigateur ira d'abord sur le bouton, puis sur Texte 1, Texte 2 et Texte 3, ce qui n'est pas forcément le comportement souhaité :

```
<div tabindex="1">Texte 1</div>
<div tabindex="3">Texte 3</div>
<div tabindex="2">Texte 2</div>
<button type="button">Mon bouton</button>
```

Pour ne pas impacter l'ordre, il est possible d'utiliser la valeur 0 pour le `tabindex`. De cette façon, le navigateur lira les éléments dans leur ordre d'apparition.

```
<div tabindex="0">Texte 1</div>
<div tabindex="0">Texte 2</div>
<div tabindex="0">Texte 3</div>
<button type="button">Mon bouton</button>
```

À noter qu'il est également possible d'utiliser la valeur -1 pour le `tabindex`. Cette valeur a 2 utilités :

- Exclure un élément de la navigation via la tabulation
- Rendre un élément focalisable sans qu'il puisse être atteint via la tabulation

```
<button id="unaccessible-btn" type="button" tabindex="-1">Mon bouton inaccessible</button>
```

Dans le cas d'une `<div>` qui ne serait pas accessible par défaut, le fait d'ajouter `tabindex="-1"` la rendra accessible via le `.focus()` mais pas via la tabulation.

- Les liens :

C'est le moyen le plus utilisé pour naviguer. Pour être compréhensible via un lecteur d'écran, aucun intitulé de lien ne doit être vide. Chaque lien doit avoir un intitulé explicite pour permettre à l'utilisateur de comprendre sa fonction et sa destination. Pour cela, on peut utiliser un attribut `<title>`.

Exemple :

```
<a href="..." title="Lire la suite de l'article">Lire la suite...</a>
```

Les liens doivent être facilement identifiables graphiquement. Par convention et par défaut, ils sont soulignés et lors d'une navigation au clavier, l'élément sur lequel porte le focus est signalé par une fine bordure en pointillés. Celle-ci permet au lecteur de se repérer et de savoir quelle action va se dérouler s'il appuie sur la touche [Entrée]. Par esthétisme, certains développeurs désactivent cet affichage mais c'est préjudiciable pour l'utilisateur navigant sans souris qui ne sait alors plus où il se situe.

Les liens ouvrant un nouvel onglet posent souvent problème. L'utilisateur en situation de handicap ne se rend pas toujours compte qu'il n'est plus sur le même site et a du mal à retourner sur la page précédente. Beaucoup de sites utilisent l'icône  pour signifier l'ouverture d'un nouvel onglet mais elle n'est pas toujours compréhensible, notamment pour les dyslexiques. Il est donc préférable d'ajouter un attribut `title = "nouvelle fenêtre"` sur le lien.

Les liens d'accès rapide permettent aussi d'accéder directement à un élément intéressant de la page. Ils sont particulièrement utiles aux utilisateurs naviguant sans souris et permettent, par exemple, aux internautes utilisant un contacteur au souffle, d'éviter de multiples efforts pour atteindre ce qu'ils cherchent. Pour que ces liens d'accès rapide restent affichés en permanence, il faut ajouter quelques lignes de code CSS :

```
.quickAccess {  
    position: absolute;  
    top: -1000px;  
}  
.quickAccess:focus {  
    top: 0px;  
    color: white;  
    background-color: black;  
}
```

Il est important également qu'un lien permette de revenir à la page d'accueil, de n'importe quelle page et en un clic. Enfin, un certain nombre de liens doivent exister pour respecter la législation : mentions légales, déclaration d'accessibilité.

- Les systèmes de navigation doivent au moins être au nombre de deux pour permettre aux visiteurs d'accéder facilement à l'information. On peut choisir parmi :
 - un menu de navigation
 - un moteur de recherche
 - un plan du site

Le moteur de recherche, le menu et les barres de navigation doivent être intégrés dans le modèle de page (template). Ainsi ils sont toujours positionnés au même endroit et accessibles de n'importe quelle page.

Le menu de navigation contient une liste de liens. Il faut donc utiliser une structure de liste avec la balise `` pour que les utilisateurs de lecteurs d'écran obtiennent la structure du menu.

Pour indiquer que le moteur de recherche n'est pas un formulaire comme les autres, il est important de lui affecter le rôle ARIA search.

```
<div role="search">
    <form method="GET" action="...">
        <input type="search" placeholder="Recherche">
        <input type="submit" value="Rechercher">
    </form>
</div>
```

Le plan du site convient pour les sites contenant peu de pages. L'information doit être bien structurée avec des titres et des listes.

Enfin le fil d'Ariane peut être une aide utile pour se repérer dans la structure du site.

Il faut également que le modèle de page contienne un moyen de contact : adresse e-mail ou lien vers un formulaire de contact.

• La structuration des pages

Les pages doivent aussi être structurées de façon logique.

Les balises de titres (`<h1>` à `<h6>`) permettent d'organiser le contenu en sections hiérarchisées. Le nom de ces balises permet de comprendre ce qu'elles contiennent et donc de structurer la page en différents blocs ayant une fonction précise. La balise `<p>` indique un paragraphe, `<q>` une citation, etc. Pour que toutes les pages soient cohérentes, on doit structurer la zone d'en-tête de chaque page via une balise `<header>`, les zones de navigation principales et secondaires via une balise `<nav>`, la zone de contenu principal via une balise `<main>` unique, la zone de pied de page via une balise `<footer>`.

Les internautes utilisant des lecteurs d'écran vont utiliser cette hiérarchie, l'historique de navigation ou la liste des onglets, pour se repérer et accéder rapidement à la partie qui les intéresse.

Le titre de la page est le premier élément vocalisé ; on peut donc n'utiliser qu'une seule balise `<h1>` correspondant à la balise `<title>` c'est-à-dire au titre principal de la page. Cela permet d'indiquer clairement le sujet de la page (et d'améliorer le référencement naturel de celle-ci).

De plus, on peut ajouter un attribut ARIA à ces balises sémantiques. Le RGAA invite à associer leur rôle à 4 balises :

```
<header role="banner">
<nav role="navigation">
<main role="main">
<footer role="contentinfo">
```

Dans chaque page web, les listes doivent aussi être correctement structurées. Pour cela, il faut utiliser :

- les balises HTML `` et `` pour les listes non ordonnées
- les balises HTML `` et `` pour les listes ordonnées

La langue utilisée pour rédiger une page doit aussi être obligatoirement renseignée. Cela est indiqué par l'attribut `lang` de la balise HTML. Cette information est primordiale pour l'accessibilité comme pour le référencement naturel puisqu'elle permet aux lecteurs d'écran d'utiliser la bonne langue (et non celle par défaut) et de prononcer le contenu avec l'accent qui convient.

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

Leur syntaxe comprend une sous-balise de langue, éventuellement complétée par une sous-balise de script. La sous-balise de langue est un code représentant la langue (ex : fr français ; eng Anglais ; de Allemand etc.) ; la sous-balise de script est composée de 4 caractères. Elle indique le système d'écriture. Par exemple, Brai correspond au braille. Ainsi fr-Brai correspond à un texte en français écrit en braille.

Les tableaux de données, possiblement présents dans une page web, sont difficiles à restituer avec un lecteur d'écran. Il faut donc éviter les tableaux complexes ou prévoir un résumé des informations transmises. Au minimum, le tableau doit avoir un titre pertinent, c'est-à-dire une légende.

```
<table>
  <caption>titre</caption>
  ...
</table>
```

Les documents à télécharger :

Sur un site, on propose souvent des documents en téléchargement. Mais quelle est l'accessibilité de ces documents ?

- Le format HTML est parfaitement accessible à condition de respecter les règles du WCAG.
- Les formats bureautiques d'Office ou leurs équivalents libres sont accessibles à condition de respecter les mêmes règles que pour un site internet : utiliser des titres, un contraste suffisant et saisir les alternatives aux images.
- Le format PDF est accessible à condition que la source ayant permis de le produire soit accessible et que la conversion effectuée conserve bien le texte en tant que texte. Un document scanné ou transformé en PDF à l'aide d'une imprimante virtuelle est complètement inaccessible.

b) En matière de design

Le design est un élément important mais il ne doit pas être essentiel à la consultation du site. Il faut séparer le fond de la forme. Le style ne doit donc pas être intégré dans une page HTML mais externalisé dans un fichier CSS et l'ordre des éléments doit rester logique même si le style CSS est désactivé.

On peut réutiliser la même feuille de style dans plusieurs pages sans duplication de code. Cela permet aux personnes malvoyantes ou dyslexiques d'adapter la page soit avec un paramétrage de leur système ou de leur navigateur, soit en ajoutant elles-mêmes des feuilles de style supplémentaires. Cela leur permet de grossir le texte, d'écarter les lignes, de changer les couleurs... Il faut s'assurer que, dans chaque page web, l'augmentation de la taille des caractères jusqu'à 200% au moins ne provoque pas de perte d'information.

- **Couleur et contraste**

Il ne faut pas se reposer uniquement sur les couleurs pour véhiculer une information, même si ces couleurs ont une forte signification (le vert et le rouge par exemple) car ceux qui ne peuvent les distinguer (ex : les daltoniens) seront mis en difficulté. Ainsi, en plus de marquer en rouge les champs obligatoires d'un formulaire, il faut les marquer aussi avec un astérisque ou expliciter la demande avec du texte lisible par un lecteur d'écran.

Lorsqu'on choisit une palette de couleurs pour un site web, on doit d'abord s'assurer que la couleur du texte (au premier plan) contraste bien avec la couleur du fond, sinon les personnes avec un handicap visuel ne pourront pas lire le contenu.

Le contraste se définit par la différence de luminance entre 2 couleurs. Cette luminance varie de 0 pour le noir à 1 pour le blanc. Le contraste est calculé selon la formule suivante :

$$\text{Contraste} = \frac{L_1 + 0,05}{L_2 + 0,05}$$

Avec L_1 = luminance de la couleur la plus claire

Et L_2 = luminance de la couleur la plus foncée

Le contraste varie entre 1 (noté 1:1) et 21 (noté 21:1)

Le RGAA conseille :

- pour les textes d'une taille inférieure ou égale à 24 px, sans gras, un rapport de contraste minimum de 4,5:1
- pour les textes d'une taille supérieure ou égale à 24 px, sans gras, un rapport de contraste minimum de 3:1
- pour les textes d'une taille inférieure ou égale à 18,5 px, avec gras, un rapport de contraste minimum de 4,5:1
- pour les textes d'une taille supérieure ou égale à 18,5 px, avec gras, un rapport de contraste minimum de 3:1

Pour atteindre le niveau AAA du WCAG en matière de contraste, celui-ci doit être de 7:1 au minimum pour le texte et de 4,5:1 pour les gros titres.

De même, le rapport de contraste des différentes couleurs composant un élément graphique doit être au minimum de 3:1. Il en est de même pour la couleur d'un lien par rapport au texte qui l'environne.

À noter : Un contraste élevé permet à toute personne, handicapée ou non, utilisant un écran brillant de mieux lire les pages dans un environnement lumineux.

On peut utiliser un outil comme [Color Contrast Checker](#) ou [Colorable](#) pour vérifier si la palette de couleurs choisie contraste suffisamment.

Mauvais exemple de contraste :



© Axopen

- **Mise en forme du texte**

La largeur du texte : pour que le texte soit facilement lisible, il faut que les lignes ne soient pas trop longues. Le critère d'accessibilité de niveau AAA recommande de ne pas dépasser 80 caractères par ligne.

La taille de la police doit pouvoir être redéfinie par l'utilisateur. L'idéal est de ne pas imposer de taille et de laisser les utilisateurs choisir celle qui leur convient le mieux, en paramétrant leur système.

Le choix de la police de caractères : La police doit être la plus lisible possible. Les personnes souffrant de dyslexie peuvent avoir des difficultés à lire un texte écrit avec une police avec empattements (serif en anglais). Il est donc préférable de choisir une police sans empattement, plus sobre et donc plus facile à lire (ex : Tahoma - avec laquelle j'ai choisi d'écrire ce mémoire -, Arial ou Verdana). Depuis peu, de nouvelles polices sont spécialement conçues pour les dyslexiques. Elles se distinguent par le tracé des lettres :

- asymétrie des lettres « b » et « d », « p » et « q »
 - distinction du « l » minuscule, du « I » majuscule et du chiffre « 1 »
 - chasse suffisante des caractères, pour éviter la confusion de « rn » avec la lettre « m »
 - de bons jambages ascendants (des b, d, f, h, k, l, t, et de toutes les capitales) et descendants (des g, j, p, q, y)
 - des lettres « g » et « a » proches de l'écriture manuscrite, etc.
- mais elles n'ont pas toutes fait la preuve de leur utilité.

Les effets de style nuisent à la bonne lisibilité du texte. Il faut donc bannir les effets d'ombre, les changements de couleur ou les mots écrits en majuscules. Pour mettre en valeur un élément essentiel du texte, il est préférable d'utiliser la balise **** qui affiche le texte avec une police en gras en indiquant qu'il s'agit d'un texte important. Cette information est alors utilisable par un moteur de recherche (ce qui n'est pas le cas avec l'utilisation de la règle CSS `font-weight` ou de la balise ``).

Enfin, le soulignement indique un lien. Il ne faut donc pas souligner à d'autres fins pour ne pas perturber les internautes.

Attention à l'usage de l'écriture inclusive ; pour le moment, les lecteurs d'écran ne sont pas capables de lire correctement ces textes et les personnes souffrant de dyslexie peuvent être gênées par cette pratique. Pour rendre un site accessible aux personnes ayant une déficience cognitive, on peut fournir une version FALC : Facile À Lire et à Comprendre. Les normes à respecter sont listées dans [L'information pour tous : règles européennes pour une information facile à lire et à comprendre](#), beaucoup recoupent les règles d'accessibilité déjà présentées. Une page web en version FALC doit être la plus épurée possible et sans animation.

Exemple de fichier falc.css :

```
.falc {  
    font-family : Tahoma, sans-serif ;  
}
```

Enfin, pour être compris, les acronymes doivent être associés, au moins une fois, à leur forme développée :

```
<p>Le contenu web est marqué à l'aide de <abbr title="Hypertext Markup  
Language">HTML</abbr>.</p>
```

Et pour faciliter la lecture par le lecteur d'écran, les abréviations doivent être explicitées (au lieu d'écrire Jan, écrire Janvier), et les tirets évités (au lieu d'écrire 5–7, écrire 5 à 7).

- **Focus sur les éléments**

Indiquer l'élément sur lequel on se trouve est très important lorsqu'on se déplace avec le clavier. Sans cela, l'utilisateur pourrait ne plus savoir où il se trouve ; d'autant plus pour les personnes souffrant de troubles de l'attention qui ont besoin d'un repère visuel.

- **Style des liens**

Les conventions de style sur les liens sont le soulignement et une couleur différente (par défaut : bleu) dans leur état normal (non visité) de celle utilisée lorsque le lien a déjà été visité (par défaut : violet) et de celle utilisée lorsque le lien est activé (par défaut : rouge). De plus, le pointeur de la souris se change en icône de pointeur lorsque les liens sont déplacés, et le lien reçoit une surbrillance lorsqu'il est ciblé (par exemple, via une tabulation) ou activé. Ces règles doivent être respectées pour ne pas perturber l'utilisateur, handicapé ou non.

- **Images et sons**

Il faut être attentif à leur poids et à l'information qu'ils apportent. Il convient aussi de s'assurer qu'ils sont correctement restitués aux utilisateurs de lecteur d'écran.

- Textes alternatifs aux images et aux sons

Si le contenu textuel est accessible via un lecteur d'écran, il n'en est pas de même pour le contenu multimédia : le contenu image/vidéo ne peut pas être vu par les personnes malvoyantes et le contenu audio ne peut pas être entendu par les malentendants.

Les textes alternatifs sont donc très importants pour l'accessibilité des personnes ayant un trouble visuel ou auditif. Il faut donc inclure dans toutes les images qui contiennent un contenu pertinent une balise `alt` apportant un texte alternatif. Il faut que celui-ci puisse être lu via un lecteur d'écran.

Il faut néanmoins distinguer deux types d'images : les images purement décoratives et celles contenant de l'information.

Les images décoratives n'apportent aucune information supplémentaire au texte. Il faut alors que la valeur de l'attribut `alt` soit vide.

```

```

Ainsi le lecteur d'écran comprend que cette image n'apporte rien d'essentiel en termes d'information et ne restitue rien.

Si possible, il faut utiliser CSS pour afficher des images qui ne sont que des décos.

Par contre, les images contenant des informations doivent pouvoir être restituées par un lecteur d'écran. Cette alternative doit contenir toutes les informations nécessaires pour comprendre l'image sans la voir. Les légendes ont leur importance (copyright, date, nom des personnes présentes sur l'image) et doivent pouvoir être lues. Il faut pour cela associer l'image à sa légende grâce aux balises `<figure>` et `<figcaption>`.

```
<figure role="figure" aria-label="la légende">
    
    <figcaption>la légende</figcaption>
</figure>
```

Les icônes purement décoratives doivent aussi être ignorées par les lecteurs d'écran. Pour cela, il faut utiliser l'attribut `aria-hidden` avec la valeur `true`. Pour celles qui sont porteuses d'informations (une loupe par exemple pour faire une recherche, il faut mettre en place une alternative en utilisant un attribut `title`, `aria-label` ou `aria-labelledby` sur une balise englobante.

```
<button aria-label= "rechercher">
    <span class= "fas fa-search" aria-hidden="true"></span>
</button>
```

- Animations

Il est possible de réaliser des animations avec du code CSS. Elles sont préférables à celles réalisées avec JavaScript car elles sont optimisées par le navigateur. Néanmoins il faut utiliser les animations avec parcimonie car elles peuvent être perturbantes pour les personnes ayant des troubles de l'attention. Les mouvements de l'écran peuvent en effet provoquer chez les personnes souffrant de troubles de l'oreille interne, des vertiges ou de la fatigue. Il faut donc se limiter aux animations réellement nécessaires. On peut aussi utiliser la media query `prefers-reduced-motion` pour ne jouer les animations que si la personne a configuré son système en ce sens.

- Médias audio et vidéo

Il est essentiel que le lecteur ait le contrôle de la lecture. La lecture ne doit donc pas être lancée automatiquement, d'une part parce que l'internaute ne souhaite pas forcément écouter ce média, d'autre part parce qu'un son déclenché sans que l'internaute ait été prévenu peut être dérangeant. Il faut donc bannir l'attribut `autoplay` des balises `<audio>`. La méthode la plus simple pour que l'utilisateur ait accès aux boutons de contrôle audio est d'utiliser l'attribut `controls` affichant le lecteur du navigateur.

Pour une bonne lisibilité, la vidéo doit être adaptée à la taille de l'écran de l'utilisateur. Comme pour les contenus audio, si la vidéo contient une partie vocale, il est nécessaire de la sous-titrer pour les personnes malentendantes. Sur son serveur web, on peut proposer une ou plusieurs pistes de sous-titres pour une vidéo. Cela se réalise grâce à la balise `<track>` dans la balise `<video>`. Il faut produire un fichier WebVTT (Web Video Text Track) contenant les sous-titres et les plages de temps durant lesquelles ils doivent s'afficher.

Une fois que le fichier.vtt est généré, il faut l'associer à la vidéo.

```
<video controls>
    <source src="..." type="video/mp4">
    <track kind="subtitles" label="nom du sous-titrage"
srclang="code de la langue" src="fichier.vtt" default>
</video>
```

c) En matière d'interactivité

Pour que l'utilisateur puisse interagir, il faut que les consignes données soient claires. Il ne faut pas utiliser un langage trop complexe, ni de jargon inutile.

En matière d'interface utilisateur, il est important d'utiliser du HTML sémantique, plutôt qu'une combinaison de CSS et de JavaScript.

- **Liens**

Les liens qui ouvrent une nouvelle fenêtre ont un effet déstabilisant pour les utilisateurs de lecteurs d'écran. Il faut donc signaler les liens qui s'ouvrent dans une nouvelle fenêtre et éviter les liens de type « cliquez ici » ou « continuer », qui sont incompréhensibles si la synthèse vocale parcourt tous les liens à la suite.

- **Boutons**

Les boutons doivent être traités de la même manière que les liens. Ils ne doivent pas être vides. Leur intitulé doit être pertinent et compréhensible. Ainsi, aux intitulés « OK », « Valider », etc. ; il faut préférer « créer un compte », « rechercher ». Les boutons doivent aussi être bien visibles. Pour éviter toute confusion, il ne faut pas mettre plusieurs boutons importants l'un à côté de l'autre et laisser un maximum d'espace autour de chaque bouton.

- **Images cliquables**

Pour une image à zone cliquable, il est important de fournir une alternative textuelle à chaque zone afin de rendre ces liens accessibles aux personnes utilisant un lecteur d'écran.

Exemple de carte de France où la région Hauts-de-France est cliquable et renvoie vers « hautsDeFrance.html ».

```

<map name="france-regions">
    <area alt="Hauts de France" title="Hauts-de-France"
          href="hautsDeFrance.html" coords="..." shape="poly">
</map>
```

- **Formulaires**

Le remplissage d'un formulaire est souvent fastidieux, d'autant plus pour les personnes en situation de handicap. De plus, si le formulaire est mal réalisé, il peut n'être pas du tout utilisable. Les libellés de formulaire sont importants puisqu'ils indiquent à l'utilisateur quelles données doivent être entrées dans chaque champ. L'objectif est que l'utilisateur saisisse correctement les informations. Il faut donc lui donner les indications nécessaires pour qu'il y parvienne facilement.

La difficulté est d'autant plus grande si l'utilisateur utilise un lecteur d'écran. Le texte positionné à côté des champs n'est pas lu s'il n'est pas associé au champ de saisie et l'utilisateur ne sait donc pas ce qu'il doit saisir.

Un formulaire se caractérise par la balise `<form>`.

Associer au champ de saisie une balise `<label>` ayant un attribut `for`, permet de rendre le formulaire plus compréhensible puisqu'elle permet d'obtenir la lecture du label lorsque

l'utilisateur arrive dans le champ. Chaque champ de formulaire doit posséder un attribut `id` dont la valeur doit être égale à celle de l'attribut `for`.

Ex : `<label for="nomFamille">Nom de famille</label>`
`<input id="nomFamille" type="text" name="nom">`

Cette formulation convient mieux que :

Remplis ton nom : `<input type="text" id="name" name="name">`

L'utilisateur doit toujours savoir à quoi correspond un champ et connaître l'ordre des champs à remplir.

Il est possible de laisser des informations accessibles uniquement aux lecteurs d'écran, en créant une classe dédiée, comme le fait le framework Bootstrap avec sa class « `sr-only` ». (« `sr` » = screen reader).

```
<a href="#">En savoir plus<span class="sr-only"> sur nos offres mobiles</span></a>
```

La mise en forme du formulaire est d'autant plus importante pour une personne ayant un champ visuel réduit. Le fait de ne pas avoir le libellé du champ collé à la zone de saisie est alors problématique. Pour remédier à ce souci, on peut aligner les libellés à droite avec la règle CSS suivante :

```
label {  
    text-align: right;  
}
```

Les champs à remplir obligatoirement doivent être marqués par l'attribut `required`. Ainsi le formulaire ne peut pas être soumis tant que les champs obligatoires n'ont pas été renseignés. Les personnes utilisant un lecteur d'écran entendent que le champ est obligatoire. Pour les autres, il est nécessaire d'avoir une indication graphique du caractère obligatoire de ce champ.

Nom *

Pour aider l'utilisateur à compléter le formulaire, il faut lui indiquer le format attendu dans les champs de saisie libre. Cela peut être fait par une indication dans le label.

```
<label for="numeroSecuriteSociale">  
    Numéro de sécurité sociale  
    <span id="formatNumeroSecuriteSociale">(13 chiffres) </span>  
    *  
</label>  
<input type="text" name="numeroSecuriteSociale"  
    id="numeroSecuriteSociale" required>
```

Dans tous les cas, les utilisateurs ont le droit à l'erreur, encore plus les personnes avec des déficiences cognitives ou motrices qui peuvent saisir des données erronées. Ils doivent pouvoir les corriger ou compléter facilement. Il ne faut donc pas que les champs soient désactivés après avoir été renseignés une première fois.

Les messages d'erreur doivent être situés au plus près des champs concernés. C'est ainsi plus simple pour l'utilisateur de procéder aux corrections.

- **Captchas**

Les captchas permettent d'éviter l'utilisation des formulaires par des robots. Mais ils sont souvent un obstacle pour certains utilisateurs : malvoyants, dyslexiques, etc. Il est donc nécessaire de prévoir une alternative non graphique comme l'envoi d'un code par sms ou e-mail. Une autre possibilité est d'utiliser l'API ReCAPTCHA de Google qui est accessible pour les personnes en situation de handicap.



© Greenspector

Les développeurs ont une grande responsabilité puisque la façon dont ils codent rejoue sur la possibilité ou non des personnes en situation de handicap à utiliser le site qu'ils ont développé. Pour s'assurer de l'accessibilité de leur site, au fur et à mesure de l'avancée du projet et avant de livrer leur travail pour la revue de code, les développeurs doivent effectuer les tests appropriés.

2) Tester et rendre un site existant plus accessible

S'il est préférable de penser à l'accessibilité dès la conception d'un site, il arrive souvent de devoir rendre plus accessible un site existant. Il faut alors faire un état des lieux. Les audits peuvent aussi être utiles au cours du développement pour vérifier les pages qui viennent d'être créées et valider le travail accompli.

a) Méthodologie de test : les principaux points à vérifier

En premier lieu, il est important de valider le code source. Il existe pour cela des outils automatisés. Comme premier test, on peut déjà désactiver le CSS du site à analyser et voir à quel point il est compréhensible sans ce dernier. Il est possible de le faire manuellement juste en retirant le CSS du code, mais la façon la plus simple reste d'utiliser les fonctionnalités du navigateur, par exemple :

- Firefox : Sélectionnez *Affichage > Style de page > Aucun style* depuis le menu principal.
- Safari : Sélectionnez *Développement > Désactiver les styles* depuis le menu principale (pour activer le menu *Développement*, choisissez *Safari > Préférences > Avancés > Montrer le menu développement dans la barre de menu*).
- Chrome : Installez l'extension Web Developer Toolbar, puis redémarrer le navigateur. Cliquez sur l'icône engrenage qui apparaîtra, puis sélectionnez *CSS > Désactiver tous les styles*.
- Edge : Sélectionnez *Vue > Style > Aucun style* depuis le menu principal.

La méthode technique la plus complète est celle du RGAA. Elle permet de mesurer le niveau d'accessibilité d'un service numérique en vérifiant que les pages web sont conformes aux 50 critères de succès des niveaux A et AA de la norme internationale WCAG 2.1 qui ont été retenus dans la norme européenne de référence.

Cette méthode propose un cadre de vérification de la conformité aux exigences d'accessibilité. Elle comporte 106 critères de contrôle. En voici quelques-uns, parmi les plus importants :

- **L'utilisation du clavier est-elle optimisée et privilégiée ?**

Les utilisateurs qui ont des problèmes de motricité fine utilisent principalement le clavier pour se déplacer sur la page. Ils doivent pouvoir effectuer leur navigation et utiliser toutes les fonctionnalités sans devoir recourir à une souris.

Voici la méthode préconisée pour vérifier la navigation au clavier :

- Positionner le curseur dans la barre d'adresse, puis parcourir toute la page avec la touche Tabulation. Et essayer de cliquer sur les liens avec la touche « Entrée » pour parcourir d'autres pages.
- Rafraîchir la page. S'assurer qu'on peut naviguer de lien en lien (ou tout autre élément focusable tel que bouton, champ de formulaire...) à l'aide de la touche tabulation. Lorsque le focus est posé sur un lien ou un bouton, la touche d'espacement active ce lien ou ce bouton. Les boutons sont également activables grâce à la touche Return.
- Utiliser aussi la combinaison de touches maj + tab pour vérifier qu'on peut naviguer à reculons.
- S'assurer que le focus est visible, c'est-à-dire qu'on distingue bien où la tabulation nous a conduits.

Il faut aussi vérifier que chaque script est contrôlable par le clavier et par tout dispositif de pointage (souris, toucher, stylet, ...)

- **Les éléments multimédias sont-ils sous-titrés et contrôlables avec le clavier ?**

- Vérifier que chaque image porteuse d'information a une alternative textuelle et que celle-ci est pertinente.
- Vérifier que chaque légende d'image est correctement reliée à l'image correspondante.
- Vérifier que chaque image de décoration, sans légende, est correctement ignorée par les technologies d'assistance.
- Vérifier que pour chaque image utilisée comme CAPTCHA, une solution d'accès alternatif au contenu ou à la fonction du CAPTCHA existe.

- Vérifier que chaque média temporel pré-enregistré (éléments <audio>, <video> ou <object>) a des sous-titres synchronisés, une transcription textuelle ou une audio-description pertinente.
- Vérifier que la consultation de chaque média est contrôlable par le clavier et tout dispositif de pointage.

Pour cela, il faut contrôler la présence des fonctionnalités obligatoires de contrôle de la consultation :

- Au minimum : lecture, pause ou stop ;
- Si le média a du son, il doit avoir une fonctionnalité d'activation / désactivation du son ;
- Si le média a des sous-titres ou une audio-description, il doit avoir une fonctionnalité de contrôle de leur apparition/disparition

Ces fonctionnalités doivent être activables par le clavier et tout dispositif de pointage. Si c'est le cas pour chaque média, le test est validé.

- Vérifier que chaque média est compatible avec les technologies d'assistance.

- **Les textes sont-ils adaptés à la vue de chacun ?**

- Dans chaque page web, le contraste entre la couleur du texte et la couleur de son arrière-plan est-il suffisamment élevé ? Pour cela, il convient de vérifier que les critères décrits dans la partie design sont respectés.
- L'information ne doit pas être donnée uniquement par la couleur. Cette règle est-elle respectée ?
- Est-ce que le ratio éléments textuels / autres éléments est respecté ? Ce ratio doit être de 5/1.
- Est-ce qu'il existe une fonctionnalité permettant de faire varier la taille du texte ? Il faut vérifier que les rapports de taille entre le titre, les sous-titres et le corps de texte se maintiennent avec le changement de dimension et que lorsque l'on agrandit la taille des textes, certaines parties ne se chevauchent pas ou ne disparaissent pas.

- **Le contenu est-il clair et bien hiérarchisé ?**

- Chaque page web doit avoir un titre de page pertinent.

Méthodologie :

Retrouver dans le document le titre structuré au moyen d'un élément <title> ;

Vérifier si le contenu de l'élément <title> est suffisamment pertinent (il permet de retrouver la page dans l'historique de navigation ou la liste des onglets).

Si c'est le cas, le test est validé.

- Dans chaque page web, la structure du document doit être cohérente.

Vérifier que la zone d'en-tête est structurée au moyen d'un élément <header> ;

Vérifier que les zones de navigation principales et secondaires sont structurées au moyen d'un élément <nav> ;

Vérifier que l'élément <nav> n'est pas utilisé en dehors de la structuration des zones de navigation principales et secondaires ;

Vérifier que la zone de contenu principal est structurée au moyen d'un élément <main> ;

Si le document possède plusieurs éléments <main>, vérifier qu'un seul de ces éléments est visible (les autres occurrences de l'élément sont pourvues d'un attribut hidden) ;

Vérifier que la zone de pied de page est structurée au moyen d'un élément <footer>.

Si c'est le cas pour chaque zone de contenu, le test est validé.

- Dans chaque page web, chaque liste doit être correctement structurée.

Retrouver dans le document les éléments regroupés visuellement sous la forme d'une liste ; pour chaque liste, vérifier que la liste est structurée :

- soit au moyen des balises HTML et pour les listes non ordonnées,
- soit au moyen des balises HTML et pour les listes ordonnées,
- soit au moyen d'éléments pourvus d'attributs WAI-ARIA role="list" et role="listitem" (quelque soit la liste).

- Dans chaque ensemble de pages, le menu et les barres de navigation doivent être toujours à la même place.
- Dans chaque page web, chaque lien doit avoir un intitulé explicite.
- Dans chaque page web, l'ordre de tabulation doit être cohérent.

Parcourir dans le document l'ensemble des contenus au moyen de la touche de tabulation vers l'avant (touche `tab`) et vers l'arrière (touches `maj+tab`),
Vérifier que l'ordre de déplacement du focus reste cohérent par rapport au contenu,

Si c'est le cas, le test est validé.

À noter : il n'est pas obligatoire que la tabulation suive l'ordre de lecture naturel (de gauche à droite et de haut en bas par exemple) tant que les éléments sont accessibles dans un ordre cohérent.

- Dans chaque page web, l'ouverture d'une nouvelle fenêtre ne doit pas être déclenchée sans action de l'utilisateur.
- **Les formulaires, les boutons et les champs sont-ils facilement accessibles ?**

Il doit être possible de se déplacer dans le formulaire et de cliquer sur les boutons (bouton « valider » par exemple) avec le clavier.

Les champs obligatoires doivent être clairement indiqués (pas uniquement par un jeu de couleurs).

Les instructions doivent être bien intégrées en haut du formulaire.

Les erreurs de remplissage du formulaire doivent être faciles à repérer (via une croix rouge par exemple).

Le RGAA déroule ainsi plus de 100 questions à vérifier point par point, ce qui prend beaucoup de temps. Cependant, pour un audit, le RGAA autorise à ne traiter qu'un échantillon de pages. Celles-ci doivent être représentatives et intégrer un document téléchargeable, une page de recherche, une page contenant un formulaire ainsi que les pages d'accueil, de contact, d'aide, d'authentification, de mentions légales, du plan du site et celle contenant la déclaration d'accessibilité. Enfin, il existe des outils de test qui permettent de gagner en efficacité.

b) Outils automatisés de test existants

- **Valider le code :**

C'est la première chose à vérifier. Le code source s'affiche en faisant un clic-droit sur la page à inspecter et en sélectionnant « Examiner l'élément », « Inspecter » ou « Code source de la page ».

Pour le **HTML**, la validation implique de s'assurer que toutes les balises sont correctement fermées et imbriquées. La sémantique HTML est normalement accessible sans modification et elle est utilisable par des technologies d'assistance comme les lecteurs d'écran.

La validation des pages HTML peut se faire grâce au [Validator](#) du W3C qui vérifie le code et retourne une liste d'erreurs

Le **CSS** est beaucoup moins accessible que HTML et ne doit être utilisé que pour le style. Pour la validation des feuilles de style CSS, [CSS Validator](#) du W3C permet de vérifier que les noms sont correctement épelés, que les valeurs des propriétés sont valides pour les propriétés auxquelles elles s'appliquent, et qu'aucune accolade ouvrante ou fermante n'a été oubliée.

Une autre option à envisager est ce qu'on appelle les applications Linter, qui peuvent avertir de mauvaises pratiques dans l'usage de CSS. Ex : [CSS Lint](#). La plupart des éditeurs de code ont leurs plugins linter. Par exemple, l'éditeur de code [Atom](#) de Github possède plusieurs plugins, avec beaucoup d'options de linting (CSS, HTML).

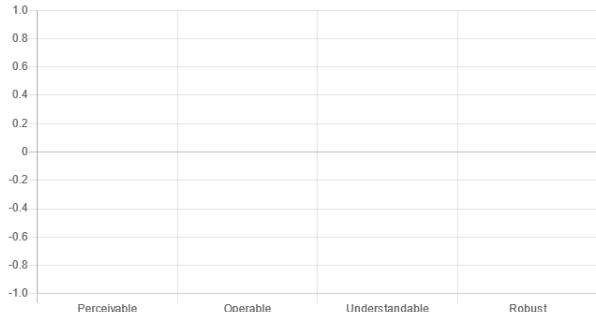
JavaScript pose le même type de problèmes que CSS concernant l'accessibilité. Il faut vérifier que les éléments <div> n'ont pas été utilisés avec du code JavaScript pour simuler une fonctionnalité et leur préférer les fonctionnalités fournies par HTML.
La validation du code JavaScript peut être effectuée avec [JSHint](#).

- **Valider l'accessibilité :**

Il existe des outils qui vérifient automatiquement la bonne mise en œuvre des pratiques d'accessibilité. Ils permettent de gagner du temps mais ne vérifient que quelques-uns des critères. Ils doivent donc être complétés.

Quelques outils d'analyse de l'accessibilité web :

- [Tanaguru](#) est une solution open source qui permet de vérifier le respect du RGAA. Pour chaque critère automatisable, les pages sont analysées et validées ou pas. Les résultats sont présentés dans un tableau de bord et les lignes posant problèmes sont clairement identifiées.



Nombre de tests 1

✗ Non conforme	0
? À qualifier	0
✗ Non testé	0
✓ Conforme	21
○ Non applicable	28

Nombre d'anomalies par thème

Perceivable	0
Operable	0
Understandable	0
Robust	0

Exemple d'un audit généré sur Tanaguru

- [Wave](#) est un outil en ligne de test d'accessibilité qui accepte une adresse web et retourne une vue annotée de la page avec les problèmes d'accessibilité surlignés.

- [Deque's aXe tool](#) est un outil d'audit qui vérifie aussi les pages et retourne des erreurs d'accessibilité. Il se distingue grâce à son extension pour navigateurs :
 - [aXe pour Chrome](#)
 - [aXe pour Firefox](#)
- [Tenon](#) est un site (dont la version complète est payante) qui traverse le code à une URL fournie et qui retourne les résultats sur les erreurs d'accessibilité en comprenant les indicateurs, des erreurs spécifiques accompagnés des critères WCAG qu'elles affectent, et des suggestions de résolutions.
- [La Va11ydette](#) d'Orange permet aussi de générer un audit de conformité d'une page web ou d'une application mobile.

Exemple ci-dessous avec l'audit de la page d'accueil du site Orange.com

Critères	Conformes	Non-conformes	Non-applicables	Taux de conformité			
Niveau	A	AA	A	AA			
Accueil	15	6	4	5	11	9	70%

Les outils de développement des navigateurs sont aussi intéressants : accessibles avec la touche [F12], ils donnent des informations sur l'accessibilité de la page consultée.

Firefox dispose d'un onglet « Accessibilité » qui permet de rechercher les problèmes de contraste, d'inaccessibilité de certains éléments au clavier et les éléments n'ayant pas d'alternative textuelle. Il est aussi possible de simuler des déficiences visuelles telles que le daltonisme.

Chrome propose l'équivalent : les DevTools, dont le plug-in Lighthouse qui permet de tester l'accessibilité d'un site ainsi que sa performance (référencement naturel).

Il existe également des plugins que l'on peut intégrer dans notre navigateur pour évaluer l'accessibilité d'une page web, comme [Equal Access Accessibility Checker](#).

Développé par IBM, ce plugin disponible sur Chrome et Firefox va scanner la page web sur laquelle on se trouve et donner un score d'accessibilité. Ce scan vérifiera par exemple que l'on a bien des labels sur chaque élément qui doivent être lus (exemple: les boutons) ou encore que les contrastes entre un texte et le fond sont bien adaptés.

Certains outils ciblent un problème particulier. Ainsi pour tester et améliorer l'accessibilité de son site aux personnes ayant des problèmes de vue, on peut utiliser [WebAIM Contrast Checker](#) qui permet de mesurer la qualité du contraste entre deux couleurs d'un site.

Il faut aussi vérifier que les documents bureautiques téléchargeables dans le site sont accessibles.

Méthodologie préconisée dans le RGAA :

- Retrouver dans le site les liens et les boutons permettant de télécharger un fichier au format bureautique ;
- Pour chaque fichier trouvé, vérifier la présence d'une version alternative présentée comme accessible :
 - Pour les documents au format .pdf, analyser le fichier avec l'outil PAC (PDF Accessibility Checker) et vérifier l'absence d'erreur d'accessibilité dans le document. À noter, cet outil ne fonctionne que sur la plateforme Windows.
 - Pour les documents au format .doc ou .docx, analyser le fichier avec le vérificateur d'accessibilité standard de Microsoft Office (à partir de la version 2010) accessible via le menu « Fichier > Informations > Vérifier la présence de problèmes > Vérifier l'accessibilité ». Ce vérificateur peut être considérablement amélioré via le plugin Word Accessibility Plug-in. Ce plugin ne fonctionne que sur Windows.
 - Pour les documents au format .odt, analyser manuellement le document avec l'éditeur OpenOffice ou enregistrer le document au format .docx et le vérifier via le vérificateur d'accessibilité de Microsoft Office 2010.
 - Pour les documents au format EPUB/DAISY, analyser le document avec un éditeur EPUB/DAISY ; l'utilitaire Ace by DAISY App permet d'effectuer le travail de validation d'un fichier EPUB 3 de manière efficace.

c) Expérience utilisateur et expertise

Pour vérifier l'accessibilité d'un site, il faut absolument le tester avec un lecteur d'écran en respectant son mode d'emploi (Voici ceux de : [VoiceOver](#) pour IOS, [NVDA](#) pour Windows).

Pour une personne malvoyante qui utilise un lecteur d'écran, une page web est une simple liste d'éléments HTML qu'il parcourt jusqu'à trouver celui qu'il cherche. Le logiciel lit à haute voix (ou retranscrit en braille) de manière linéaire chaque nouvel élément textuel qu'il rencontre.

Lors du test, il faut vérifier que :

- Le lecteur d'écran lit à voix haute chaque élément au fur et à mesure de la progression dans le contenu, en notifiant ce qui est un paragraphe, ce qui est un titre, etc.
- Il s'arrête après chaque élément et permet la navigation.
- Il est possible de dresser une liste de tous les titres afin de les utiliser comme une table des matières utile pour trouver un contenu spécifique.

Si le code est illogique, le lecteur d'écran n'a pas d'indicateur, il ne peut pas récupérer une table des matières utilisable, et la page entière est vue comme un bloc unique, lu d'une traite.

Il peut être très intéressant de faire tester son site par une personne en situation de handicap qui maîtrise bien les outils d'assistance comme les lecteurs d'écran ou les plages braille. La DINUM (Direction interministérielle du numérique) a créé un groupe d'utilisateurs handicapés pour tester l'accessibilité des sites et applications mobiles publics. Il inclut actuellement une trentaine d'utilisateurs, majoritairement handicapés visuels, une dizaine de personnes handicapées cognitives et quelques handicapés moteurs. Ces utilisateurs ne sont pas des experts en accessibilité, mais ont pour point commun d'avoir l'habitude de naviguer sur internet et de posséder les matériels et logiciels nécessaires à cette fin. Si les tests effectués ne remplacent ni ceux réalisés à l'aide des logiciels de test, ni ceux réalisés par des professionnels experts en accessibilité, ils peuvent avoir, néanmoins, une grande utilité pour montrer le degré d'efficacité des publics concernés sur les sites et applications testés.

On peut aussi faire appel à des plateformes qui mettent en relation avec des testeurs. Ceux-ci testent le site en donnant leur avis en direct sur les éléments que l'on souhaite évaluer. Ce type d'évaluation permet de connaître l'avis d'internautes plus ou moins ciblés sur l'accessibilité d'un site. Mais ces testeurs ne sont pas des professionnels.

Il est aussi possible de demander l'avis d'experts en accessibilité web.

La plupart des experts se basent sur la méthodologie [WCAG-EM](#) (Website Accessibility Conformance Evaluation Methodology), qui comprend 5 étapes :

- La définition du périmètre d'évaluation. Que s'agit-il d'évaluer ? Quels sont les objectifs de l'évaluation (atteindre le niveau A, AA, AAA...) ?
- L'exploration du site. Quelles sont les pages principales ? Quelles sont les fonctionnalités clés ? Quels sont les différents types de contenus proposés ? Quelles sont les technologies requises ?

- La sélection d'un échantillon représentatif, s'il n'est pas possible d'évaluer chaque page.
- L'audit de l'échantillon. Il s'agit de s'assurer que les critères d'accessibilité sont remplis et d'identifier les problèmes.
- Le rapport d'audit, qui rassemble les conclusions de l'audit et calcule un score global d'accessibilité.

d) Corrections et déclaration de l'accessibilité

Une fois le site évalué, il faut réaliser les corrections pour en améliorer l'accessibilité. Celles-ci peuvent être priorisées en fonction de leur importance, d'autant plus s'il s'agit d'un gros site.

On peut traiter prioritairement :

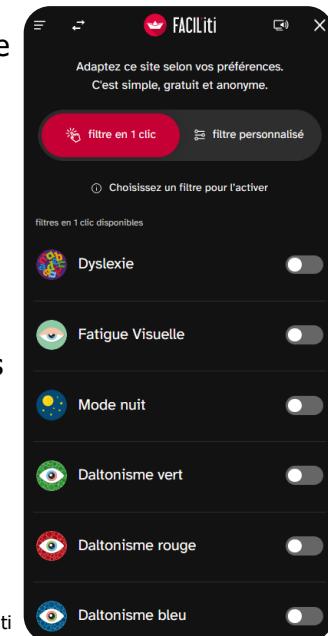
- les contenus clés et ceux qui sont les plus consultés,
- les pages interactives,
- les problèmes d'accessibilité remontés par les internautes,
- les nouveaux contenus et fonctionnalités à intégrer sur le site.

On peut affiner ces règles en se focalisant d'abord :

- sur les problèmes faciles, rapides et peu coûteux à corriger,
- sur les problèmes qui ont le plus d'impact pour l'usager. En particulier les points liés à la navigation, à la page d'accueil ou au processus d'achat (soit les problèmes qui correspondent au niveau A du référentiel WCAG 2.0).

En première intention, pour améliorer rapidement le confort de navigation en ligne, on peut utiliser une extension de navigateur comme [Confort+](#). Mis au point par Orange, elle offre une vingtaine d'options pour adapter un site web : augmenter la taille des polices, accentuer le contraste des couleurs, modifier les mises en page ou naviguer par pointage. Un seul paramétrage suffit pour que tous les sites web prennent en compte les préférences de l'utilisateur. Disponible en OpenSource, Confort+ est téléchargeable gratuitement sur les navigateurs Firefox, Chrome et Internet Explorer.

Dans le même registre, [Facil'iti](#) permet d'adapter l'affichage des sites web aux besoins des utilisateurs.



Dans le cadre d'une refonte, il peut être pertinent de revoir entièrement le design plutôt que de réparer un à un les problèmes.

Une fois l'audit terminé, il faut remplir la déclaration d'accessibilité en respectant le format imposé.

La déclaration doit contenir les informations suivantes :

- un état de conformité :
 - Conformité totale : si tous les critères de contrôle du RGAA sont respectés
 - Conformité partielle : si au moins 50 % des critères de contrôle du RGAA sont respectés
 - Non-conformité : s'il n'existe aucun résultat d'audit en cours de validité permettant de mesurer le respect des critères ou si moins de 50 % des critères de contrôle du RGAA sont respectés
- un signalement des contenus non accessibles
- des dispositifs d'assistance et de contact pour permettre à quiconque de signaler tout défaut d'accessibilité et à une personne handicapée de demander une solution alternative accessible
- la mention de la faculté pour la personne concernée de saisir le Défenseur des droits, en cas d'absence de réponse ou de solution.

La déclaration d'accessibilité est valide à partir de sa date de publication. Elle doit être mise à jour :

- à la date de modification substantielle ou de refonte du site concerné,
- 3 ans après la date de publication de la déclaration ou 18 mois après la date de publication d'une nouvelle version du référentiel.

Pour les sites internet, la déclaration d'accessibilité doit être publiée dans un format précis et accessible sur le site internet concerné. Elle est mise à disposition au sein d'une page « accessibilité », directement accessible depuis la page d'accueil et depuis n'importe quelle page du site.

V) Quid de l'accessibilité des applications mobiles ?

1) L'accessibilité sur IOS et Androïd

De nos jours, les appareils mobiles peuvent gérer des sites web complets et la grande majorité des téléphones incluent de base des outils permettant de rendre l'interface accessible à tous. Les principales plates-formes ont même des lecteurs d'écran intégrés : VoiceOver sur IOS et TalkBack sur Androïd.

Ce ne sont pas les seuls outils. Il est souvent possible de gérer la taille du texte pour les personnes malvoyantes, la colorimétrie pour les personnes atteintes de daltonisme et avec [Axopen](#) pour IOS, on peut même être notifié lors de la détection de certains types de sons (alarme, aboiement, ...) pour les personnes atteintes de surdité.

Tous ces outils rendent la majorité des applications plus accessibles et ne nécessitent pas d'intervention de la part des développeurs.

En matière de développement, pour rendre un site utilisable sur mobile, il convient de suivre les bonnes pratiques en matière de conception et d'accessibilité (comme expliqué dans la partie IV).

Par exemple, il faut s'assurer que le contraste est suffisant entre les différents éléments (textes, images...) et leur arrière-plan. Il convient aussi d'ajouter des sous-titres et de l'audiodescription pour les éléments sonores et les vidéos.

Il faut également soigner l'User eXperience (UX) et l'User Interface (UI) afin d'avoir une application claire et structurée, avec peu d'éléments par écran et des boutons de grande taille aux actions évidentes. Chaque icône doit avoir un texte alternatif. Cela permet au lecteur d'écran de communiquer verbalement à l'utilisateur à quoi elle correspond.

Spécifiquement, pour le menu de navigation, généralement réduit à une petite icône ≡ appelée "menu hamburger", placée en haut de l'écran, il faut s'assurer que le contrôle qui le révèle est accessible et que le reste de la page est déplacé ou caché pendant l'accès au menu, afin d'éviter toute confusion lors de la navigation.

Ces actions permettent non seulement de pallier certains handicaps visuels et cognitifs mais aussi de rendre l'application plus agréable pour les personnes valides.

Néanmoins, il y a des éléments spécifiques à prendre en compte. Sur les appareils mobiles, la saisie de données a tendance à être plus difficile que sur les ordinateurs en raison de la petitesse du clavier. Il vaut donc la peine de minimiser la quantité de frappe nécessaire et de prévoir des modalités de saisie multiples (texte, voix).

Voir ces guides sur les critères de conception [IOS / Android](#) ou [Guidelines For Mobile Web Development](#) pour plus d'informations.

2) Le responsive design

Le responsive design permet que les mises en page et autres fonctionnalités des applications mobiles changent de manière dynamique en fonction de facteurs tels que la taille de l'écran et la résolution, de sorte qu'elles soient utilisables et accessibles aux utilisateurs de différents types d'appareils.

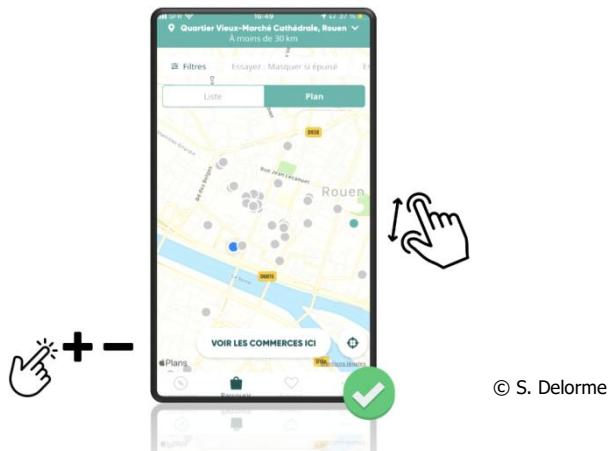
Les problèmes les plus courants auxquels le développeur d'applications mobiles doit faire face sont les suivants :

- les mises en page. Par exemple, une mise en page à plusieurs colonnes ne fonctionnera pas aussi bien sur un écran étroit et il faudra peut-être augmenter la taille du texte pour le rendre lisible. Ces problèmes peuvent être résolus en créant une mise en page réactive utilisant des technologies telles que [media queries](#), [viewport](#) et [flexbox](#).
- la taille des images téléchargées. En général, les appareils de petite taille n'ont pas besoin d'images aussi volumineuses que les ordinateurs de bureau et ils risquent davantage d'être sur des connexions réseau lentes. Il est donc sage de proposer des images plus petites sur des dispositifs à écran étroit. On peut gérer cela en utilisant [responsive image techniques](#). Il faut néanmoins penser aux écrans haute résolution qui ont besoin d'images de résolution supérieure pour que l'affichage reste net. À noter : de nombreuses exigences en matière d'images peuvent être satisfaites grâce au format d'images vectorielles SVG, bien pris en charge par les navigateurs actuels. SVG a une petite taille de fichier et restera net quelle que soit la taille affichée (voir [Adding vector graphics to the web](#) pour plus de détails).

- Les changements d'orientation. Il est essentiel de permettre à l'utilisateur de choisir l'orientation qu'il souhaite pour afficher une application. Certaines personnes ayant des difficultés motrices peuvent devoir changer l'orientation de leur smartphone pour limiter leur fatigue. D'autres ne peuvent le tenir en main ni modifier son orientation. C'est, par exemple, le cas des personnes en situation de handicap moteur qui fixent leurs appareils à un bras articulé et le contrôlent à l'aide de contacteurs. Si l'application n'est pas capable de s'adapter à l'orientation de l'appareil, l'expérience utilisateur sera très fortement dégradée.



- le focus. Il faut permettre à l'utilisateur de zoomer sur les différents éléments de l'application, en s'assurant que les éléments ne sont pas tronqués lorsqu'ils sont agrandis. De plus, il faut intégrer que de nombreux utilisateurs peuvent être dans l'incapacité de réaliser certains mouvements sur l'écran. Il est donc essentiel de prévoir des alternatives aux fonctionnalités qui nécessitent des gestes complexes. Si l'action de zoomer/dézoomer sur une carte se fait en pinçant l'écran, il faut également qu'il soit possible de le faire en utilisant des boutons.



- les zones cliquables. Pour qu'une personne affectée de tremblements ou empêchée de pointer avec précision (déformations articulaires) puisse cliquer sur une zone, il faut que celle-ci soit suffisamment large. Il est recommandé que chaque élément d'UI interactif dispose d'une zone sélectionnable, ou zone cible tactile, d'au moins 48 dp × 48 dp. (dp = Pixel indépendant des appareils. Si le dp est utilisé en tant qu'unité de longueur, le système d'exploitation de l'appareil mappe la valeur dp vers un nombre de pixels correspondant basé sur la résolution de l'écran de l'appareil. Pour ce mappage, 1 dp est considéré être égal à 1 pixel sur un écran de résolution 160 dpi. Le nombre correspondant des pixels peut être calculé avec la formule px = dp * (dpi/160). En utilisant des pixels indépendants de l'appareil, on obtient donc un meilleur contrôle du design sur plusieurs appareils à résolutions variées.)
Pour les utilisateurs de dispositifs de pointage adapté, l'étiquette d'un champ permet d'étendre la surface de clic et améliore ainsi l'efficacité des manipulations.

L'extrait de code suivant montre un élément avec la taille de zone cible tactile recommandée :

```
<ImageButton ...  
    android:paddingLeft="4dp"  
    android:minWidth="40dp"  
    android:paddingRight="4dp"  
  
    android:paddingTop="8dp"  
    android:minHeight="32dp"  
    android:paddingBottom="8dp" />
```

Les applications mobiles sont, elles aussi, concernées par l'obligation de conformité au RGAA mais la méthode technique applicable est celle de la directive européenne. On peut tester l'accessibilité d'une application grâce à des tests automatisés [sous Android](#) ou [sous IOS](#)

Enfin, la déclaration d'accessibilité doit être disponible sur le site internet de l'organisme qui a développé l'application ou apparaître avec d'autres informations disponibles lors du téléchargement de l'application. L'état de conformité est précisé au sein de l'application.

3) Des exemples réussis d'aide à l'autonomie

Voici quelques applications mobiles accessibles, développées pour améliorer l'autonomie des personnes en situation de handicap, qui ont fait parler d'elles :

Pour les personnes à mobilité réduite :

- [J'accède](#) est une plateforme collaborative sur laquelle chacun peut renseigner l'accessibilité d'un lieu. Téléchargeable sur Android et IOS et consultable à partir d'un navigateur internet sur n'importe quel terminal (ordinateurs, smartphones, tablettes...), elle informe gratuitement sur l'accessibilité de tous les lieux ouverts au public (plus de 119 000 établissements référencés).
- [Parking](#) est l'une des parties de l'application Handicap.fr. Comme son nom l'indique, elle permet de trouver des places de parking PMR à proximité de la localisation / destination recherchée.

Pour les aveugles et malvoyants :

- [Soundscape](#): ce projet de recherche initié par Microsoft aide les personnes malvoyantes à se repérer dans l'espace grâce aux sons.

Pour les sourds et malentendants :

- [RogerVoice](#) est une application qui sous-titre les appels téléphoniques. À chaque appel, elle se branche au réseau pour analyser et retranscrire ce que l'interlocuteur dit.
- [Ava](#) présente aussi un dispositif de retranscription assez complet. L'application peut accompagner les malentendants dans le cadre professionnel ou scolaire en retranscrivant par écrit les propos tenus lors des réunions ou cours en ligne.

Pour les personnes ayant des difficultés à s'exprimer

- [Proloquo2go](#) est une application complète pour les personnes ayant du mal à s'exprimer. Elle propose plus de 28 000 pictogrammes classés par sujets (lettres, couleurs, actions, métiers, émotions) pour pouvoir communiquer.
- [Helpicto](#) est une application qui aide, notamment les autistes, à communiquer en remplaçant les mots par des images. Deux personnes peuvent ainsi échanger via l'application, en sélectionnant des pictogrammes. Elle est aussi utile en cas de surdité, etc.
- [Grid](#) est une application de Communication Alternative Améliorée (CAA) qui permet aux utilisateurs de vocaliser des phrases (à écrire ou à préenregistrer), de contrôler leur domotique ou d'accéder à leur PC (Windows) via une fenêtre de contrôle d'ordinateur utilisable avec la commande oculaire ou un simple contacteur et un système de défilement d'icônes en icônes. Cet outil convient donc particulièrement aux personnes polyhandicapées ou atteintes de paralysie cérébrale.

Conclusion

En guise de conclusion, j'aimerais rappeler que l'accessibilité web n'est pas seulement bénéfique aux personnes souffrant de handicap ; elle profite à tout le monde.

Les services numériques représentent une réelle opportunité d'inclusion dans la société des personnes en situation de handicap. Pour cela, il faut que ces services soient conçus et développés selon des règles précises. Il reste encore beaucoup de chemin à parcourir puisque seuls 4% des sites internet en France sont pleinement accessibles, et beaucoup demeurent inaccessibles, dont de nombreux sites gouvernementaux. En effet, d'après une circulaire interministérielle de 2020, si tous les services de communication au public sur le Web, fournis par l'État ou les collectivités territoriales, sont soumis à l'obligation d'accessibilité depuis 2012, « seules 13 % des 250 démarches administratives en ligne les plus utilisées par les Français prennent en compte l'accessibilité. La Commission européenne place la France au 19e rang sur les 27 pays de l'UE pour l'accessibilité de ses services publics en ligne. ». Toutefois, la législation évolue et une prise de conscience est en cours, comme en témoignent les mesures annoncées par le gouvernement lors de la Conférence Nationale du Handicap du 26 avril 2023 qui promet un plan de rattrapage pour l'accessibilité numérique, afin de garantir d'ici 3 ans, l'accessibilité des démarches et sites internet publics. Côté entreprises, le bilan n'est pas meilleur. On constate un manque de volonté des décideurs pour faire en sorte que l'accessibilité soit prise en compte au moment de la refonte ou de la réalisation des sites internet alors que l'accessibilité numérique constitue un levier important : inclusion sociale, meilleur référencement, élargissement de l'audience et rentabilité accrue. Si la loi de 2005 et le RGAA existent, le contrôle de l'application de la loi n'est pas suffisamment exercé.

La sensibilisation doit aussi se faire auprès des étudiants et des acteurs de la filière numérique. Selon le rapport du Conseil National du Numérique, presqu'un développeur sur 2 n'a jamais entendu parler de l'accessibilité numérique ! Comme la sécurité ou le respect des données personnelles, l'accessibilité numérique doit donc faire partie intégrante des enseignements, dans les cursus de formation (initiale et continue) aux métiers du Web.

Bibliographie et sitographie

Boisgontier, Hervé : Green IT et accessibilité : développez votre site web numérique responsable.- Saint-Herblain : ENI (Ressources informatiques), 2021.

[Andrea studio](#)

[Android](#)

[Article extrait du journal Les Echos \(septembre 2021\)](#)

[Article extrait du magasine Usine Digitale \(mars 2017\)](#)

[Axopen](#)

[Cartographie numérique du handicap](#)

[Cellenza](#)

[Centre national de l'expertise](#)

[Conseil national du numérique](#)

[Fabrique du net](#)

[Fédération des aveugles de France](#)

[Handicap.fr](#)

[Insee](#)

[Ipredis](#)

[Observatoire du numérique](#)

[Organisation Mondiale de la santé](#)

[RGAA](#)

[TechLab](#)

[Usabilis](#)