

# Choisir et utiliser un CMS pour créer des contenus accessibles

Livre Blanc – Septembre 2015

Coordonné par **Olivier NOURRY**

## Remerciements

Ce livre blanc reprend des contributions faites le 25 juin 2015 à la Cité des Sciences à Paris lors du 21<sup>ème</sup> Séminaire du Groupe de Travail AccessiWeb (GTA). Ce séminaire a été réalisé avec le soutien de la Cité des Sciences, d'Océane Consulting (Tanaguru), de Temesis, de V-Technologies ([Partenaires Premium](#)) et d'Urbilog (Partenaire).



## A propos de BrailleNet

L'association BrailleNet est une association loi 1901 à but non lucratif. Elle œuvre depuis 1997 pour l'accessibilité numérique en faveur de toutes les personnes qui, du fait d'un handicap, ont des difficultés pour utiliser les services et outils numériques.

BrailleNet a développé des ressources et services pour la mise en œuvre effective de l'accessibilité numérique des sites web (création du référentiel AccessiWeb; organisation de formations à l'accessibilité numérique à destination des contributeurs, rédacteurs et développeurs; labellisation de sites web). Elle coordonne et initie des projets de recherche en collaboration avec ses partenaires scientifiques (INRIA, Université Pierre et Marie Curie, l'Institut de la Vision, l'INSERM) et avec des organismes privés.

BrailleNet organise chaque année des séminaires, conférences, ainsi que le Forum Européen de l'Accessibilité Numérique à la Cité des Sciences et de l'Industrie. Elle propose des services en faveur des personnes handicapées, telle que la Bibliothèque Numérique Francophone Accessible qui offre plus de 30 000 ouvrages dans différents formats adaptés.

En 2003 l'association BrailleNet décide de créer le Groupe de Travail AccessiWeb afin de développer l'expertise dans le domaine des technologies de l'accessibilité numérique (techniques, méthodologies, ...). Ce groupe contribue notamment à augmenter le nombre de professionnels du Web capables de réaliser des services numériques accessibles.

Conformément aux statuts et aux missions de l'association, l'activité du Groupe de Travail AccessiWeb est sans but lucratif. Il est avant tout basé sur le volontariat. Ses travaux ne concernent pas les aspects commerciaux du marché de l'accessibilité numérique.

### Coordinateur

Olivier NOURRY (Smile)

### Editeurs

Dominique BURGER (UPMC-INSERM, BrailleNet) & Katie DURAND (BrailleNet)

## Table des matières

<b>Préface</b> / Dominique Burger, UPMC-INSERM, BrailleNet .....	3
<b>Introduction</b> / Olivier Nourry, Smile .....	5
<b>Chapitre 1 : Le cadre normatif</b> .....	6
<i>Présentation d'ATAG 2.0, les recommandations du W3C pour l'accessibilité des CMS</i> / Jean-Pierre Villain, Access42 .....	6
<b>Chapitre 2 : La politique d'accessibilité de trois grands CMS</b> .....	10
<i>Adapter un CMS pour répondre au besoin - retour d'expérience avec eZ Publish</i> / Christophe Caron, Telmedia* .....	10
<i>L'accessibilité à grande échelle : Comment WordPress intègre l'accessibilité à son processus de développement</i> / Olivier Nourry, Smile .....	16
<i>Des solutions accessibles grâce à Drupal</i> / Mike Gifford, OpenConcept .....	21
<b>Chapitre 3 : L'amélioration de l'accessibilité d'un CMS</b> .....	25
<i>L'accessibilité via les thèmes enfants dans WordPress</i> / Gaël Poupard, Kosmos .....	25
<i>L'accessibilité dans le monde Joomla !</i> / Ariane Andurand, C3RB Informatique .....	28
<b>Chapitre 4 : Etude comparative de la capacité des CMS à produire des contenus accessibles</b> Olivier Nourry, Smile ; Claire Bizingre, Accesbilis ; Edouard Cunibil, Happyculture ScopARL ; Jacques Pyrat, Pyrat.net ; Mickael Ruzafa, C3rb informatique.....	30
<b>Glossaire</b> .....	37
<b>Les partenaires</b> .....	39

Ce document est disponible en format PDF balisé à l'adresse suivante :

[http://www.accessiweb.org/tl\\_files/doc\\_telechargement/gta21/choisir-utiliser-un-CMS.pdf](http://www.accessiweb.org/tl_files/doc_telechargement/gta21/choisir-utiliser-un-CMS.pdf)

## Préface

Dominique BURGER

Il est aujourd'hui acquis que l'accessibilité doit être traitée en amont d'un projet Web. Cela implique en particulier de faire le choix, dès l'origine, d'un « bon » Système de Gestion de Contenu ou CMS (Content Management System).

Le CMS c'est ce que [W3C](#) désigne du terme plus générique d'*Authoring Tool*. [W3C/WAI](#) soulignent son importance et l'interdépendance entre les différents composants qui contribuent à l'accessibilité finale des contenus (figure 1). Afin de guider le travail des développeurs, [W3C/WAI](#) a élaboré trois types de normes, [ATAG](#), [WCAG](#) et [UAAG](#), concernant respectivement les outils de gestion de contenus (*authoring tools*), les contenus Web eux-mêmes (*contents*) et les navigateurs (*user agents*).

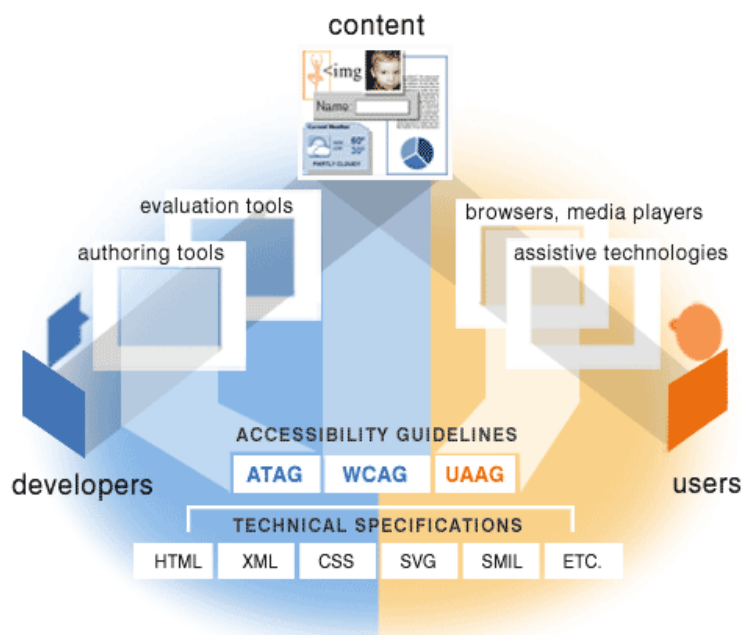


Figure 1 : Les composants essentiels de l'accessibilité Web (source [W3C/WAI](#) – <http://www.w3.org/WAI/intro/components.php>)

BrailleNet, partenaire du W3C depuis 1999 et membre depuis 2010, a suivi les travaux de WAI sur ATAG. En 2010 nous avons invité Jeanne SPELLMAN<sup>1</sup> à présenter le plan d'action de W3C pour l'élaboration de la norme ATAG 2.0, conséquence de la publication de WCAG2.0 en 2008. Après cinq ans de travaux, cette norme est publiée (automne 2015).

BrailleNet a souhaité traiter cette question, lors du 21<sup>ème</sup> séminaire AccessiWeb, d'un point de vue pratique, avec des développeurs confrontés quotidiennement à la question de l'accessibilité.

Avec l'aide d'Olivier NOURRY, expert AccessiWeb confirmé et membre actif du GTA depuis 2006, nous avons fait appel à des développeurs experts de différents CMS afin de comparer leur prise en compte

<sup>1</sup> Conférence de Jeanne SPELLMANN 2<sup>ème</sup> Forum Européen de l'Accessibilité Numérique :  
[http://www.dailymotion.com/video/xda638\\_jeanne-spellman-1-2-4e-forum-europe\\_tech](http://www.dailymotion.com/video/xda638_jeanne-spellman-1-2-4e-forum-europe_tech) (partie 1),  
[http://www.dailymotion.com/video/xda6pr\\_jeanne-spellman-2-2-4e-forum-europe\\_tech](http://www.dailymotion.com/video/xda6pr_jeanne-spellman-2-2-4e-forum-europe_tech) (partie 2)

de l'accessibilité. Cela a permis de produire une étude inédite présentée dans ce livre blanc. Cette étude détaille les points forts et faibles de quatre CMS largement répandus et résume leur aptitude à produire et gérer des sites Web accessibles. Ces résultats seront utiles aux développeurs, aux chefs de projet, et à tout responsable amené à s'interroger sur la stratégie à mettre en œuvre pour améliorer l'accessibilité d'un service en ligne et le choix d'outils appropriés. Cette étude propose aussi une méthode qui permettra de faire évoluer cette base de connaissances sur l'accessibilité.

BrailleNet avec ses partenaires publics et privés, sont mobilisés pour faire avancer l'accessibilité numérique en France. Ils se réjouissent du rôle unique joué par le réseau des 500 membres du Groupe de travail AccessiWeb pour faire progresser cette expertise et diffuser les bonnes pratiques et les connaissances qui en découlent.

C'est la raison pour laquelle BrailleNet a décidé de prolonger ce séminaire en publiant ce livre blanc. L'ambition de BrailleNet dans le futur sera de poursuivre cette étude, en l'actualisant et en l'étendant à d'autres CMS.



Dominique BURGER  
Ingénieur de Recherche  
INSERM - Institut de la Vision  
Université Pierre et Marie Curie



## Introduction



### Olivier NOURRY, Consultant et Formateur Accessibilité - Smile

Professionnel du Web depuis 2000, consultant et formateur en accessibilité du Web depuis 2006. Riche d'une expérience en développement, en gestion de projet et en assistance MOE et MOA, Olivier accompagne les organisations dans la mise en œuvre de l'accessibilité au sein de leurs projets de création ou refonte de sites Web. Il forme développeurs, responsables de projet et rédacteurs à l'application des recommandations du [RGAA](#) et des bonnes pratiques.

La question de l'accessibilité des CMS (Content Management System, Système de Gestion de Contenu, en anglais) est une question récurrente, généralement sous la forme « Quel CMS choisir pour faire un site accessible ? », ou encore « Comment réaliser un site accessible avec le CMS dont je dispose ? ».

Ces questions n'ont pas de réponse universelle. Un CMS est généralement un objet logiciel complexe et plusieurs facteurs de choix entrent en ligne de compte. Chaque réponse est spécifique : le contexte, les objectifs, les capacités des utilisateurs, les adaptations réalisées, modifient substantiellement les stratégies et techniques qu'il convient d'appliquer.

A la base du besoin d'un CMS, on trouve le principe de séparation des tâches, entre les domaines rédactionnel et technique. Un CMS permet de :

- Simplifier un processus complexe et répétitif ;
- Se concentrer sur le contenu ;
- Réduire la dépendance rédacteur de contenus par rapport aux techniciens.

Environ un site web sur deux est administré par l'intermédiaire d'un CMS. À partir d'un certain niveau de complexité du site, il est probable que cette proportion augmente, jusqu'à devenir largement dominante. La capacité des CMS à produire et administrer des contenus dans le respect des règles d'accessibilité est donc un enjeu majeur.

L'accessibilité des CMS, si l'on se réfère au standard [ATAG 2.0](#)<sup>2</sup> du W3C, s'évalue à trois niveaux :

- Accessibilité des interfaces de contribution (*back-office*)
- Accompagnement de l'utilisateur dans la création de contenus accessibles
- Accessibilité des contenus mis en ligne, et consultables par les internautes (*front-office*)

Lors du 21<sup>ème</sup> Séminaire Technique du Groupe de Travail AccessiWeb (GTA21), consacré à l'accessibilité des CMS, nous avons cherché à apporter des réponses aux questions que se posent les professionnels du Web, chargés soit de publier des contenus accessibles en ligne, soit de mettre à disposition les outils permettant de le faire. Ce document a été réalisé sur la base des travaux présentés à cette occasion, et en reprend la ligne directrice :

- Présentation du cadre normatif : le standard ATAG2
- La politique d'accessibilité de trois grands CMS : eZ Publish, WordPress et Drupal
- L'amélioration de l'accessibilité d'un CMS, par modification du noyau ou par ajout de thèmes
- Etude comparative de l'accessibilité de quatre CMS du point de vue des contributeurs

<sup>2</sup> <http://www.w3.org/TR/ATAG20/>

# Chapitre 1 : Le cadre normatif

## Présentation d'ATAG 2.0, les recommandations du W3C pour l'accessibilité des CMS



Jean-Pierre Villain - Access42

Jean Pierre Villain est en charge du Pôle R&D de la société Access42, dont il est co-fondateur. Auparavant, il avait travaillé en tant que Senior Accessibility Expert chez TecAccess, aux États-Unis, où il était en charge de rédiger une version opérationnelle des normes internationales pour ses clients et a été fondateur de Qelios. Principal rédacteur du référentiel AccessiWeb coordonné par l'association BrailleNet depuis la version 1.1, Jean-Pierre est également l'artisan du référentiel technique du [RGAA 3.0](#). Il a été formateur des Experts AccessiWeb en Évaluation depuis 2007 pour le compte de l'association BrailleNet. En dehors des référentiels, il voue un amour passionné à Javascript et CSS.3

### En guise d'introduction

Authoring Tool Accessibility Guidelines ([ATAG](#)) est une norme du [W3C](#) destinée à fournir un cadre de spécification technique aux développeurs d'outils de création de contenus Web, afin que ces outils permettent et facilitent la production de contenus accessibles. Une première version de cette norme, ATAG 1.0 a été publiée en 2000. La version en cours, ATAG 2.0, est cohérente avec la norme [WCAG](#) 2.0.

### Quelques caractéristiques d'ATAG 2.0

- ATAG 2.0 est le fruit d'un travail commencé en 2005
- C'est un document assez court, dense et très technique
- Sa structure est similaire à celle des WCAG : une partie contenant des règles et critères et un guide d'implémentation technique
- ATAG 2.0 s'applique aux technologies Web (type CMS) et non-Web (type logiciel), ce qui est une des raisons de sa complexité
- Cette version est en *Proposed Recommendation* W3C et sera publiée Septembre 2015

### Structure d'ATAG 2.0

- ATAG 2.0 est structuré en deux grandes parties:

Partie A : faire en sorte que l'outil de création soit lui-même accessible

Partie B : accompagner la production de contenus accessibles

- ATAG 2.0 fixe trois niveaux de priorités (A, AA, AAA)

### Portée de la norme ATAG 2.0

- Tous les types de contenus produits par l'auteur ou l'outil
- Tous les outils, en technologie Web et non-Web

## Partie A : faire en sorte que l'outil de création soit lui-même accessible

L'interface de l'outil de création doit être conforme à WCAG 2.0 en technologies Web.

L'interface de l'outil de création doit être compatible avec l'accessibilité, via une série de critères spécifiques, pour les technologies non-Web.

### Fonctionnalités spécifiques

#### Niveau A

- Si l'interface de visualisation est personnalisable, les fonctionnalités de personnalisation ne doivent pas modifier le contenu édité.
- L'interface, ses composants et ses fonctionnalités d'accessibilité doivent être documentés (via une aide ou directement dans l'interface).
- Les actions d'édition doivent pouvoir être annulées ou un mécanisme de confirmation doit être proposé.
- Si une fonctionnalité de prévisualisation est fournie, elle doit utiliser un agent-utilisateur (*user agent*) accessible du marché ou être conforme avec les User Agent Accessibility Guidelines ([UAAG](#)) niveau A.
- La durée d'une session ne doit pas être limitée ou bien il faut fournir une fonctionnalité de sauvegarde automatique régulière.

#### Niveau AA

- Si une fonctionnalité de visualisation du code est fournie, l'auteur doit pouvoir la parcourir et l'éditer.
- L'auteur doit pouvoir effectuer une recherche dans le contenu édité, y compris dans les alternatives.
- L'auteur doit pouvoir sauvegarder la configuration personnalisée de l'interface.
- L'interface, ses composants et ses fonctionnalités doivent être documentés (via une aide ou directement dans l'interface).

#### Niveau AAA

- Si une fonctionnalité de prévisualisation est fournie, l'auteur doit pouvoir choisir l'agent-utilisateur utilisé.
- Un historique des modifications doit être proposé et celles-ci doivent pouvoir être annulées.
- Une méthode pour implémenter ou modifier des raccourcis-clavier doit être fournie pour chaque fonctionnalité (ex. menus, options de menu etc.).
- L'auteur doit disposer de fonctions de navigation entre des éléments liés programmatiquement (ex. atteindre un élément par son id, par une classe de style, trouver une méthode à partir de son appel...).



## Partie B : créer des contenus accessibles

Le CMS doit permettre de :

### Niveau A

- S'assurer que les contenus produits automatiquement par l'outil sont conformes aux Web Content Accessibility Guidelines ([WCAG 2.0](#)) ;
- S'assurer que les contenus pré-édités (par un plug-in par exemple), fournis par l'outil, sont conformes aux WCAG 2.0 ;
- Faire en sorte que les processus de production d'alternatives automatiques soient contrôlables par l'auteur ;
- Alerter et assister l'auteur sur la production de contenus accessibles, ou l'existence potentielle de problèmes, lors de l'édition ou la mise à jour des contenus ;
- Proposer une fonctionnalité de vérification de l'accessibilité avant la validation du contenu.
- Assister l'auteur dans la réparation des problèmes d'accessibilité via des recommandations de correction ou des propositions de corrections automatiques ;
- Activer les options d'accessibilité par défaut et alerter du risque potentiel en cas de désactivation par l'auteur ;
- Proposer des exemples de contenus accessibles dans les dispositifs d'aides ;

### Niveau AA

- Ne pas générer ou réutiliser des alternatives sans une confirmation de l'auteur ;
- Préserver les informations d'accessibilité lors des opérations de conversion ou d'optimisation de format ou fournir un mécanisme pour rendre conformes les contenus convertis ou alerter l'auteur ;
- Présenter les fonctionnalités dédiées à l'accessibilité au même niveau que celles qui ne le sont pas ;
- Proposer les propriétés d'accessibilités prévues pour l'élément en cours d'édition ;
- Indiquer si les [templates](#) ou les contenus pré-formatés, fournis par l'outil, supportent l'accessibilité ou pas ;
- Produire un résumé de la vérification d'accessibilité.

### Niveau AAA

- Fournir un mécanisme de sauvegarde et de réutilisation des alternatives aux contenus non-textuels pour un même contenu ;
- Ne proposer que des [templates](#) conformes WCAG AA ;
- Produire des tutoriaux pour produire des contenus accessibles.

## Du rêve à la réalité

Il n'existe aujourd'hui aucun CMS totalement conforme à ATAG 2.0. En effet l'implémentation d'ATAG est complexe et très couteuse en investissement. Cependant le sujet intéresse de plus en plus de développeurs depuis le début des années 2010 ; des efforts sont faits ; des travaux sont en cours...

Récemment les vérificateurs d'accessibilité sont devenus une valeur ajoutée importante des *Rich Text Editors* ([RTE](#)).

## Ressources

- [ATAG 2.0 \(en anglais\).](#)
- [AChecker, un vérificateur d'accessibilité pour TinyMce](#)
- [Démonstration du vérificateur d'accessibilité de CKEditor](#)

## Chapitre 2 : La politique d'accessibilité de trois grands CMS

### Adapter un CMS pour répondre au besoin - retour d'expérience avec eZ Publish



**Christophe CARON, Développeur Web - Telmedia\***

Développeur/Intégrateur Web, spécialisé dans le CMS eZ Publish, Christophe travaille au sein de l'agence Telmedia\* sur des sites internet de collectivités (mairies, communautés de communes et d régions) sous eZ Publish. Il travaille sur l'accessibilité de ces projets d'un point de vue développement et intégration : mise en place de solutions pour l'accessibilité, adaptation du code HTML.

#### Les chiffres et le contexte

Le site de la région [pasdecalais.fr](http://pasdecalais.fr) est propulsé par le CMS [eZ Publish](#) depuis Janvier 2009. Ce site cache derrière lui une dizaine de sites satellites sur la même instance eZ Publish ce qui a comme principal intérêt la mutualisation de fonctionnalités telles que formulaires accessibles très souples, annuaires, etc.

L'équipe d'administration de ces sites se compose de six super administrateurs (ayant accès à l'ensemble des sites) et une quinzaine de rédacteurs spécifiques aux sites satellites.

L'ensemble du site contient environ 4000 articles et une centaine de formulaires.

L'équipe interne de [pasdecalais.fr](http://pasdecalais.fr)<sup>3</sup>, dans sa volonté de mettre en œuvre l'accessibilité, a fait le choix d'eZ Publish et s'en est montrée satisfaite.

Dans une volonté d'amélioration de l'accessibilité (entre autres), [Telmedia\\*](#) a pris en charge la refonte technique du site en 2013 sous eZ Publish (version 5).

Dans cet article nous tentons de répondre à la question : Que permet au juste ce CMS ?

#### Le CMS eZ Publish

L'intérêt principal d'eZ Publish est de fournir une interface de gestion pour l'ensemble des contenus d'un site Web sans une ligne de code. Avec eZ Publish on peut créer tous types de contenus dont on a besoin, sur mesure, juste avec des paramétrages dans l'interface de gestion, y compris :

- Une infinité de champs de différents types de données (de la simple ligne de texte, à l'éditeur WYSIWYG<sup>4</sup> en passant par le champ de géolocalisation) ;
- La possibilité de réviser (on peut reprendre une précédente version d'un contenu), déplacer, contrôler par les droits d'édition de son choix et rendre multi-langue chaque contenu ;
- Le multi-placement (on peut placer un même contenu à différents endroits du site et n'avoir à le modifier qu'une seule fois) ;

<sup>3</sup> Sous la responsabilité de Bertrand BINOIS

<sup>4</sup> [What you see is what you get](#)

- Un moteur de recherche puissant et étendu, où on sélectionne soi-même quel champ on souhaite indexer et où l'on peut combiner facilement plusieurs critères de recherche ;
- Un moteur de recherche extensible via Apache Solr<sup>5</sup> qui permet d'en étendre les possibilités (recherche géo spatiale, pertinence, suggestions de recherche, etc.) ;
- Des performances accrues avec plusieurs niveaux de caches, couplable à Varnish<sup>6</sup>, des compilations des feuilles de styles et du JavaScript ;
- Un éditeur WYSIWYG totalement personnalisable : on crée facilement des boutons pour insérer des nouveaux types de contenu.

## Objectifs du projet

Les principaux objectifs du projet de refonte du site Web du Pas de Calais sont de :

- L'adapter aux nouveaux périphériques : mettre en place une nouvelle charte graphique utilisant le *Responsive Web Design* et en HTML 5 ;
- Améliorer son administration et ses fonctionnalités ;
- Monter en force en termes d'accessibilité afin de répondre aux nouveaux critères des nouvelles certifications (RGAA 3.0) ;
- Utiliser une version à jour d'eZ Publish; La version 5 utilisant désormais le Framework PHP Symfony<sup>7</sup>.

## Les acteurs et la gestion du projet

Le découpage du projet s'est fait ainsi :

- L'équipe pasdecalais.fr livre la charte graphique ;
- Telmedia\* assure l'intégration HTML et le développement sous eZ Publish ;
- Temesis<sup>8</sup> est en charge de l'audit d'accessibilité sur un panel de pages du site internet ;
- L'ensemble des échanges autour du projet se déroule chez Telmedia\*, via l'outil de bug tracking Redmine ;
- Les travaux d'accessibilité se déroulent après l'intégration de la charte graphique et en plusieurs salves.

---

<sup>5</sup> **Solr** (prononcé "solar") est une plateforme logicielle de moteur de recherche s'appuyant sur la librairie de recherche Lucene, créée par la Fondation Apache et distribuée et conçue sous licence libre. *Solr* utilise le langage Java et est exécuté par un conteneur de servlets, comme Tomcat. Il communique avec le client à l'aide d'une interface de programmation en XML et JSON, généralement via le protocole HTTP. (Source Wikipédia)

<sup>6</sup> **Varnish** est un serveur de cache HTTP apparu en 2006 et distribué sous licence BSD. (Source Wikipédia)

<sup>7</sup> **Symfony** est un framework Modèle-vue-contrôleur(MVC) libre écrit en PHP 5 destiné majoritairement aux professionnels du développement. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web. (Source Wikipédia)

<sup>8</sup> [www.temesis.com](http://www.temesis.com)

## Objectif de conformité avec le référentiel AccessiWeb HTML5-ARIA et le RGAA 3

Pour se mettre en conformité avec le [RGAA 3.0](#) et [AccessiWeb HTML5-ARIA](#), le site a dû s'adapter à un ensemble de nouvelles bonnes pratiques:

- Aide à la navigation via [HTML5-ARIA](#) notamment dans les formulaires et les fonctionnalités faisant appel à JavaScript (menu accordéon, fenêtre modale, galerie photos)<sup>9</sup>
- Modification de termes ; ajout d'aide à la saisie sur les formulaires<sup>10</sup>
- Transformation des liens en boutons ; amélioration de la visibilité des liens au sein du contenu<sup>11</sup>
- Amélioration de l'expérience pour les lecteurs d'écran (texte caché)
- Adaptation des balises et attributs obsolètes depuis HTML5
- Ajout de fonctionnalités propres à l'accessibilité :
  - Police adaptée à la dyslexie
  - Contraste
  - Gestion de la taille de police
- Rendre la Web TV<sup>12</sup> totalement accessible en ajoutant :
  - L'audio description,
  - Le sous titrage
  - Un choix dans la couleur du sous titrage, sa taille et sa couleur d'arrière-plan

### Quelques cas concrets dans eZ Publish

A) La balise <acronym> n'existe plus en HTML5 et ne devrait plus être utilisée. Il faut utiliser l'élément <abbr> à la place.

Cette modification nécessite de revoir tous les champs du site comprenant cette balise.

B) La balise <caption> à ajouter à tous les tableaux du site en plus l'attribut summary.

C) L'ajout de nouvelles fonctionnalités au lecteur vidéo dans l'interface de gestion :

- Audio description (fichier MP3, OGG)
- Un fichier de sous-titres

Pour cela nous utilisons notamment les possibilités de l'éditeur [WYSIWYG](#) propre à eZ Publish.

---

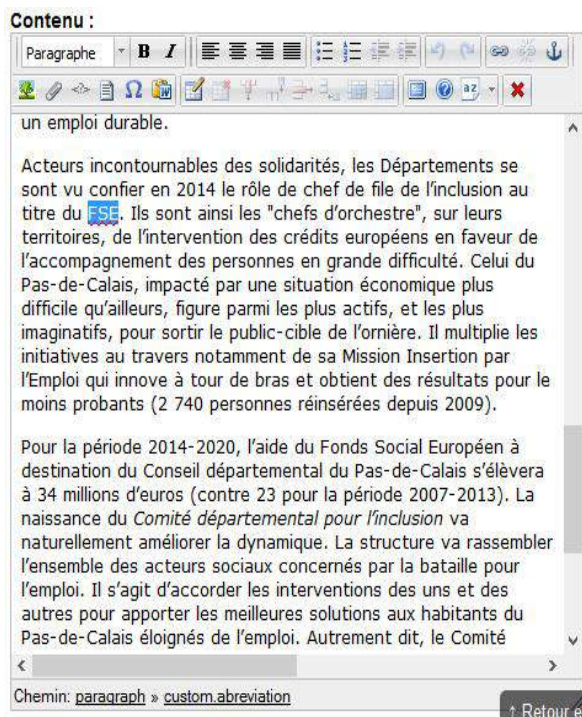
<sup>9</sup> Exemple sur le site : <http://www.pasdecalais.fr/L-institution/Vos-elus> Aide à la réalisation d'une navigation avec du html-5 ARIA : <http://www.w3.org/TR/wai-aria-practices/#accordion>  
<http://blog.telmedia.fr/accessibilite-outils-javascript-accessibles-site/>

<sup>10</sup> Exemple : <http://www.pasdecalais.fr/Formulaire-de-contact-Guidage-Personnalise-des-Sollicitations>

<sup>11</sup> Exemple de mise en avant des liens: <http://www.pasdecalais.fr/Solidarite-Sante/Enfance-et-famille/La-Protection-Maternelle-et-Infantile/Vous-attendez-un-enfant2>

<sup>12</sup> Exemple de la WEB TV : <http://www.pasdecalais.fr/Pas-de-Calais-TV/Weo-presente-Margaux-au-Parc-de-loisirs-d-Olhain> Script utilisé : <https://github.com/access42/MFPvideoPlayer>

## Un éditeur WYSIWYG, oui, mais pas n'importe lequel !



eZ Publish a un avantage majeur pour la maintenance du code HTML généré par l'éditeur. Il enregistre toutes les données de l'éditeur en base de données au format XML et non HTML. Ces données XML sont ensuite transformées par un préprocesseur où le développeur peut personnaliser le rendu HTML de chaque balise et ce par le biais d'un fichier de *template* (.tpl).

Ce rendu HTML est commun à l'ensemble des pages, on peut donc modifier à posteriori une balise utilisée par un administrateur sans qu'il ait besoin de réécrire tous les contenus. Cela reste transparent pour les rédacteurs ; le WYSIWYG est en effet semblable en ergonomie à ceux des autres CMS.

Ce principe est résumé avec le schéma suivant :

### Cas A : Remplacer la balise <acronym>

```
acronyme.tpl
1 <acronym{if ne($title|trim, '')} title="{ $title}"{/if}{
  if ne($langue|trim, '')} lang="{ $language}"{/if}>
  { $content}</acronym>
```

La balise <acronym> est affichée sur le site via un fichier de *template* (.tpl).

```
acronyme.tpl
1 <abbr{if ne($title|trim, '')} title="{ $title}"{/if}{if
  ne($langue|trim, '')} lang="{ $language}"{/if}>
  { $content}</abbr>
```

En modifiant ce fichier la balise <acronym> est ainsi modifiée en <abbr> sur l'ensemble du site.

## Cas B : La balise <caption>, personnalisation de l'éditeur de texte

Pour répondre à notre problématique de champ inexistant on peut facilement insérer ce nouveau champ dans l'éditeur. Pour cela on édite deux fichiers de configuration (.ini).

```

1 <?php /* #?ini charset="utf8"?
2
3 [CustomAttribute_table_caption]
4 Name=Caption (WAI)
5 Type=text
6 Required=false
  
```

```

[table]
CustomAttributes[]=summary
CustomAttributes[]=caption
  
```

On ajoute la nouvelle balise <caption> au fichier tpl des tableaux du site.

```

1 <table{if ne($description|trim, '')} summary=
  "{$description}"{/if} class="{classification|wash}">
  {if ne($caption|trim, '')}<caption>{$caption}</caption>
  {/if}{$rows}</table>
  
```

Grâce à l'éditeur extensible on ajoute de nouvelles propriétés à notre CMS pour répondre aux problématiques d'accessibilité. Cette balise est disponible sur l'ensemble des contenus du site.

Nouvel élément <table>

Propriétés

**Nouvel élément <table>**

Taille:

Colonnes:  Lignes:

Largeur:  %

Contour:  px

Classe:

Titre:

Description:

Summary (WAI):

Caption:

OK Annuler

## Cas C : La Web TV

Pour la Web TV on peut ajouter à tous les contenus de type vidéo autant de champs que nécessaire. On ajoute ici les sous-titres et l'audiodescription via deux champs de type fichier.

Il suffit ensuite au développeur d'ajouter ces deux nouveaux champs au fichier TPL pour étendre les possibilités de la vidéo

The screenshot shows a CMS interface with a top toolbar containing buttons for 'OK', 'Appliquer', 'Annuler', a dropdown menu set to 'Ligne de texte', and 'Ajouter l'attribut'. Below the toolbar, two file fields are visible:

- 16. Sous-titres [Fichier] (id:2456)**:
  - Nom: Sous-titres
  - Identifiant: captions
  - Description: (empty)
  - Options:  Obligatoire,  Interrogeable,  Collecteur d'information,  Désactiver la traduction
  - Par défaut (Content) ▼ Catégorie
  - Taille maximum de fichier: 0 MB
- 17. Audio description [Fichier] (id:2691)**:
  - Nom: Audio description
  - Identifiant: audio\_desc
  - Description: Audio description de la vidéo, affiche un bouton sur le site permettant de l'activer

A 'Retour en haut' button is located at the bottom right of the interface.

Dans cette capture d'écran on peut voir en haut à gauche une barre d'action qui permet d'ajouter des champs à notre contenu vidéo, ici en pleine page nous avons le détail des deux nouveaux champs.

L'avantage principal d'eZ Publish est que le développeur n'a pas eu à réaliser de script de gestion pour ces deux nouveaux champs. eZ Publish va gérer pour lui toutes les options autour de ces champs en administration. Il peut donc se consacrer au développement de la partie visible à l'internaute.

## Conclusion

eZ Publish ne propose pas nativement des interfaces accessibles aux internautes. Cependant, grâce à sa grande souplesse il est facile à adapter aux besoins en accessibilité. Le site pasdecalais.fr a obtenu le premier label e-accessible<sup>13</sup> suite aux travaux menés avec Telmedia\* et Temesis.

<sup>13</sup> Label e-accessible vise à valoriser les administrations ayant entrepris une démarche de prise en compte de l'accessibilité conforme au [RGAA 3.0](#).



## L'accessibilité à grande échelle : Comment WordPress intègre l'accessibilité à son processus de développement



### Olivier NOURRY, Consultant et Formateur Accessibilité - Smile

Professionnel du Web depuis 2000, consultant et formateur en accessibilité du Web depuis 2006. Riche d'une expérience en développement, en gestion de projet et en assistance MOE et MOA, Olivier accompagne les organisations dans la mise en œuvre de l'accessibilité au sein de leurs projets de création ou refonte de sites Web. Il forme développeurs, responsables de projet et rédacteurs à l'application des recommandations du [RGAA](#) et des bonnes pratiques.

### Contexte

On estime que WordPress est utilisé pour administrer 24% des sites visibles sur le Web, et 60% des sites gérés via un CMS. Permettre aux utilisateurs de WordPress de créer des contenus accessibles est donc un enjeu majeur pour l'accessibilité du Web. Le présent article décrit l'historique et la méthode de prise en compte de l'accessibilité dans le [core](#) (noyau) de WordPress.

Cette présentation est fortement inspirée de celle effectuée par Joe DOLSON<sup>14 15</sup> en mars 2015, en anglais. Rappelons que Joe DOLSON est le principal animateur du groupe de travail sur l'accessibilité dans WordPress, contributeur du *core* de WordPress, et du groupe de travail [Make WordPress Accessible](#)<sup>16</sup>, et auteur de [thèmes](#) et [plug-in](#) accessibles, dont *WordPress Accessibility* et *Access Monitor*.

### Historique de la prise en compte de l'accessibilité dans WordPress

L'initiative *Make WordPress Accessible* a été lancée en mars 2011. En mai 2011 les premières demandes d'évolutions pour l'accessibilité concernaient WordPress 3.2 et le thème par défaut, *Twenty Eleven*.

Jusqu'alors, bien qu'il y ait eu des initiatives individuelles (thèmes accessibles, signalement ou réparation de problèmes d'accessibilité), il n'y avait pas eu d'effort concerté pour développer des interfaces accessibles, informer, ou faire des tests utilisateurs.

De mai à novembre 2011: rien de significatif, ou presque, ne s'est passé. Après énormément de retours durant le mois de mai, l'équipe accessibilité avait été très tranquille. Mel PEDLEY<sup>17</sup>, le team leader, postait occasionnellement, mais dans le vide.

### Monter une organisation

Démarrage lent, donc... Mel PEDLEY avait porté haut le message de l'accessibilité, pendant des années. Mais on manquait de monde travaillant à la fois sur WordPress et sur l'accessibilité. On avait un leadership; mais pas d'implication. Il fallait également un « processus d'évolution ».

<sup>14</sup> Voir son site Web : [www.joedolson.com](http://www.joedolson.com)

<sup>15</sup> <http://www.slideshare.net/joedolson/massively-maintained-accessibility-wordpress>

<sup>16</sup> <https://make.wordpress.org/accessibility/>

<sup>17</sup> Voir son site [Black Widow Web Design](http://blackwidowwebdesign.co.uk/) : <http://blackwidowwebdesign.co.uk/>

## Le processus d'évolution de WordPress

Pour déterminer comment améliorer l'accessibilité d'un produit en évolution permanente comme WordPress, il faut comprendre son processus d'évolution, et s'y adapter. Voici les grandes étapes de ce processus :

- Proposer une amélioration, une correction, ou une fonctionnalité ;
- Obtenir l'adhésion d'autres développeurs;
- Fournir un feedback sur les anomalies ;
- ... Arrive ce qui doit arriver... ;
- Intégrer au *core*.

Des experts accessibilité se sont impliqués, et c'était précieux, mais leurs noms n'étaient pas reconnus par la communauté WordPress. Il fallait impliquer le *Release Lead*, personne qui définit les priorités et oriente les développements de la version à venir. Impliquer le *Release Lead* est vital.

Drew JAYNES, *Release Lead* sur WordPress 4.2, a priorisé l'accessibilité, ce qui a permis de la faire progresser énormément.

## L'Accessibilité implique de s'impliquer...

En permanence il y a plus de 300 tickets actifs<sup>18</sup> sur [Make WordPress Accessible](#) (ticket ayant connu une activité au cours des deux dernières semaines).

- Nécessite un dialogue ;
- Nécessite une implication très tôt ;
- Nécessite des gens qui fournissent des correctifs ;
- Nécessite des gens qui ont accès à la gestion de Track (*bug tracker* de WordPress).

## Combien de contributeurs?

Sur la version 3.8 on a pu compter 188 contributeurs du *core*, 267 sur la version 3.9, 275 sur la version 4.0, 283 sur la version 4.1 !

Cette importante quantité de contributeurs et de correctifs et évolutions constituent autant d'occasions d'introduire des problèmes d'accessibilité... ou leurs solutions.

## Nécessité d'informer, former les développeurs WordPress

C'est une perte de temps de seulement regarder la liste des problèmes d'accessibilité et de dire « à corriger », ou d'ouvrir une anomalie du type: « ne respecte pas le standard ». En revanche, les actions suivantes permettent de transférer la compétence, et favorisent la prise en compte de l'accessibilité dans les développements :

- Conférences aux WordCamps ;
- Articles sur Make WordPress Accessible et ailleurs ;
- Des ressources (code) ;
- Formations en ligne ;

---

<sup>18</sup> Lien vers les tickets actifs : <https://core.trac.wordpress.org/tickets/active>

- Implication active dans le suivi des tickets dans Track (système de suivi des demandes mis en place pour WordPress).

Par expérience, les stratégies efficaces consistent à :

- Etre spécifique (précis dans les demandes et solutions cibles) ;
- Prioriser les demandes ;
- Suivre leur évolution dans le système de gestion des demandes.

L'adhésion des développeurs du *core* à la prise en compte de l'accessibilité a été un succès total... (ce qui ne signifie pas pour autant qu'il y ait un accord parfait sur chacun des points).

## Où en est-on ?

Un [groupe de tests dédiés à l'accessibilité](#)<sup>19</sup>, géré par Rian RIETVELD<sup>20</sup>, valide les développements.

Pour chaque *release*, Joe DOLSON établit deux listes d'objectifs d'accessibilité. La première, en début de cycle, inclut tout ce qui doit être traité en amont, à grande échelle. La seconde est établie pour la version beta de la *release*. Elle comporte généralement des éléments plus simples à mettre en œuvre à ce stade, et plus gérables ; elle comporte aussi des optimisations sur le travail déjà réalisé.

Des demandes de consultation sur l'accessibilité sont émises par l'équipe de développement du *core*, l'équipe UX, et les développeurs de *plug-ins* de fonctionnalités. Auparavant, les nouveautés pouvaient être incluses au *core* sans validation accessibilité. Désormais, il y a des demandes pour cela. Un problème majeur d'accessibilité peut être un motif de refus de livraison.

Une bibliothèque de modèles accessibles ([WordPress Accessibility Pattern Library](#)<sup>21</sup>) est mise à disposition des développeurs.

Des tests et formations sur l'accessibilité des thèmes sont disponibles également.

## Stratégie à long terme

L'accessibilité dans WordPress connaît une évolution lente mais continue. Trois *releases* sont effectuées par an, avec des itérations individuelles. Des bibliothèques de solutions ([Let WP Speak](#)<sup>22</sup>, [WP Pattern Library](#)<sup>23</sup>) et des supports de formation/information des développeurs sont disponibles.

---

<sup>19</sup> <https://make.wordpress.org/accessibility/testing/>

<sup>20</sup> Voir site [RRWD web development](http://www.rianrietveld.com/) : <http://www.rianrietveld.com/>

<sup>21</sup> Site Web : <https://make.wordpress.org/accessibility/2015/01/15/wordpress-theme-pattern-library-for-accessibility/>

<sup>22</sup> Site Web : <https://make.wordpress.org/accessibility/2015/04/15/let-wordpress-speak-new-in-wordpress-4-2/>

<sup>23</sup> Site Web : <https://make.wordpress.org/accessibility/tag/theme-pattern-library/>

## Le problème de la rétrocompatibilité

Gérer la compatibilité de l'[API](#) pour 36 000 [plug-ins](#) and 3 000 [thèmes](#) a de nombreuses implications:

- API de paramétrage;
- Fonctions et widgets hérités d'anciennes versions ;
- Utilisation de classes CSS "pour lecteurs d'écran" ;
- Comportement des formulaires ;
- Dans l'Admin (interface d'administration), titres de sections et structure HTML ;

Et on ne parle que de ce que l'équipe peut passer en revue !

## A l'avenir

On attend des avancées majeures dans le futur, concernant notamment :

- JSON REST API<sup>24</sup>
- *Image Flow* (module de retouche d'images intégré à l'interface d'administration)

Autant de menaces que d'opportunités !...

## L'accessibilité de WordPress face à Drupal

Parmi les grands CMS, Drupal est probablement le plus accessible. Aux États-Unis, il est bien implémenté sur les sites du gouvernement fédéral et les sites liés à l'enseignement supérieur, où l'accessibilité est une exigence bien identifiée.

Cela ne signifie pas pour autant que les sites réalisés avec Drupal sont accessibles, et que ceux avec WordPress ne le sont pas.

Les choix effectués par les développeurs ont un impact sur le résultat. Et ils s'imposent toujours par rapport au comportement du [core](#).

Par exemple, dans WordPress, contrairement à Drupal, il n'y a pas de module de création de formulaire dans le *core*. Leur création est dévolue à des *plug-ins* tiers, par définition moins bien contrôlés en termes d'accessibilité.

Ce choix permet aussi de conserver un nombre restreint de fonctionnalités dans le *core*.

---

<sup>24</sup> <https://wordpress.org/plugins/json-rest-api/>

## Détecter les problèmes d'accessibilité et les résoudre

Les CMS produisent du HTML. Le HTML (valide) est accessible. JavaScript, CSS, le HTML invalide, les contenus inaccessibles altèrent l'accessibilité.

Les anomalies d'accessibilité dans WordPress peuvent avoir plusieurs origines :

- Le *core*
- Un *plug-in*
- Un *thème*

Quelques pistes pour trouver l'origine d'un problème :

- Si c'est dans l'interface d'administration : probablement dans le *core* (sauf si c'est la page de paramétrage d'un thème ou d'un *plug-in*...)
- Si c'est sur le « *front* » (visible par les internautes) :
  - Dans le menu, ou le rendu de l'article? Probablement le thème.
  - Dans un formulaire de contact, une fonctionnalité particulière de type calendrier ou un service e-Commerce: c'est un *plug-in*...

Les thèmes sur WordPress.org doivent suivre des [règles](#)<sup>25</sup> ...sauf pour les thèmes commerciaux. Les thèmes commerciaux ont leurs propres « règles ».

Pour signaler des anomalies d'accessibilité dans WordPress, il est possible de [créer un nouveau ticket](#)<sup>26</sup>. Avant de reporter quoique ce soit, il sera conseillé de tester avec tous les *plug-ins* désactivés, et avec le thème par défaut.

---

<sup>25</sup> <https://make.wordpress.org/themes/handbook/review/>

<sup>26</sup> <https://core.trac.wordpress.org/newticket>

## Des solutions accessibles grâce à Drupal



### Mike GIFFORD, Fondateur et Président - OpenConcept (Canada)

Mike a fondé OpenConcept Consulting Inc en 1999. Depuis 2000, il a été particulièrement actif dans le développement et l'amélioration de CMS open source pour permettre aux gens de se consacrer de plus près aux contenus. OpenConcept travaille avec des organismes aux profils très variés, ce qui les met dans une position unique pour bâtir une communauté et diffuser des connaissances accumulées dans une variété de secteurs. Avant de lancer OpenConcept, Mike a travaillé pour plusieurs ONG, parmi lesquelles Oxfam Canada et les Amis de la Terre.

## Drupal en bref

### Qu'est-ce que Drupal?

Drupal est un CMS open-source écrit en langage PHP. D'après [w3techs](#), 2,1% des sites Web dans le monde utilisent Drupal, et ce CMS détient la troisième plus grosse part de marché des systèmes de gestion de contenu.

Drupal est un CMS reconnu pour sa flexibilité, mais aussi pour le niveau de sécurité et d'accessibilité de son code, ainsi que ses capacités multilingues.

Drupal est utilisé dans de multiples applications telles que :

- Les sites Web, les intranets, les extranets et les portails
- Les plateformes de e-Commerce
- Les plateformes d'apprentissage en ligne (Learning Management System, ou LMS)
- Les solutions de gestion de contacts et d'utilisateurs
- Etc.

### Venez pour le logiciel, restez pour la communauté

Comme pour tout logiciel open-source, Drupal est développé et maintenu par une communauté d'utilisateurs. En Juin 2015, la communauté comptait 38 695 développeurs et 1,2 millions de comptes étaient ouverts sur drupal.org où l'on trouvait 30 973 modules et 2 135 thèmes. En moyenne, les développeurs effectuent plus de 2 000 contributions par semaine et la communauté organise plus de 2 400 événements par an à travers le monde.

Plus d'un million de sites Web utilisent Drupal aujourd'hui, et 24% de ces sites sont des sites publics.

Open-source par définition, le code de Drupal est libre d'accès et distribué sous licence GPL<sup>27</sup>. Chacun est libre de contribuer à la communauté de [développeurs](#) et cela renforce une culture dite de *leadership by humility*.

<sup>27</sup> La GNU General Public License (communément abrégé GNU GPL, voire simplement « GPL »), est une licence qui fixe les conditions légales de distribution des logiciels libres du projet GNU. Cette licence a depuis été adoptée, en tant que document définissant le mode d'utilisation, donc d'usage et de diffusion, par de nombreux auteurs de logiciels libres, en dehors des projets GNU.

## Un petit « cœur »

Le cœur de Drupal (*core* en anglais) est un fichier compressé de 3.56 Mbytes. Les versions les plus récentes (datant de juin 2015) sont les versions 7.38 et 6.36.

Le cœur de Drupal est constitué de modules nécessaires à la création d'un site basique. Pour ajouter des fonctionnalités, il faut y intégrer différents modules ou *APIs* afin d'obtenir le produit final désiré. Ce processus d'intégration fait toute la flexibilité de Drupal, et permet dans le cas des APIs de normaliser le code.

## Un processus de développement et de résolution des bugs qui a fait ses preuves

Drupal supporte deux versions stables ainsi qu'une version en développement en même temps. Au mois de Juin 2015, les deux versions stables sont Drupal 6 et Drupal 7. Drupal 8 est en développement.

Les développeurs qui contribuent à l'amélioration du code et à la résolution des bugs gèrent les problèmes au moyen d'une file d'attente publique (*issue queue*). Des *sprints* de développement sont également organisés lors des événements de la communauté (conférence ou *Developer Days*) afin de résoudre certains problèmes en groupes. De manière générale, les bugs sont résolus en priorité dans la version en développement.

La file d'attente de Drupal suit le fonctionnement suivant :

1. Un bug est signalé par un membre de la communauté ;
2. Un *patch* est développé avec les modifications ;
3. Le code est soumis à une série de tests automatisés ;
4. Des développeurs effectuent une révision manuelle et ajoutent des impressions d'écran ;
5. Le *patch* est marqué "Testé et Approuvé Par la Communauté" (*RTBC*) ;
6. Les Mainteneurs officiels valident et appliquent le *patch*.

## Comment Drupal est devenu accessible

### Le souci de respecter les standards

Drupal est reconnu pour son respect des normes, et l'accessibilité n'échappe pas à la règle. Toutefois, cela n'a pas toujours été le cas : lors de la sortie de Drupal 6, plusieurs problèmes d'accessibilité ont été identifiés, mais, à l'époque, personne n'a voulu prendre la responsabilité de les régler. Un peu plus tard, en 2008, la version 2.0 de *WCAG* est sortie, en même temps que le développement de Drupal 7 commençait. C'est alors que *Webchik*<sup>28</sup>, *Angela Byron*, *Drupal Core Committer*, a encouragé *Mike GIFFORD*<sup>29</sup> à prendre en charge l'accessibilité du code de Drupal.

---

<sup>28</sup> Voir le site : <https://twitter.com/webchick>

<sup>29</sup> Voir le site : <https://twitter.com/mgifford>

## Les améliorations de Drupal 7

Les principales améliorations effectuées pour Drupal 7 concernent les modules et fonctions suivantes, incluses dans le cœur même de Drupal (et applicable à tous les sites utilisant ce CMS) :

- Amélioration de *Form API*
- Amélioration de la fonction *CSS display:none*
- Amélioration de la fonction *Drag'n'Drop* (glisser-déposer)
- Ajouts de la fonction *Skip Links* (Sauts de navigation)
- Amélioration de la gestion des images
- Amélioration des contrastes de couleurs

La mise à jour et la création de nouveaux modules a permis de bénéficier de nouveaux outils.

L'accessibilité est devenue à tel point critique pour Drupal que la sortie de Drupal 7 a été retardée afin d'englober toutes les améliorations prévues. Le guide de Drupal 7 inclut également tout un chapitre sur l'Accessibilité.

## Drupal 8

Drupal 8 est en cours de développement et la communauté de développeurs a définis des *Core Gates*<sup>30</sup> qui peuvent retarder sa date de sortie (pour l'instant évaluée à fin 2015) :

- Documentation
- Performance
- Accessibilité
- Ergonomie
- Tests

Certains critères d'[ARIA](#) et d'HTML5 n'ont toutefois pas été inclus dans le cœur de Drupal.

Une des principales améliorations de Drupal 8 est l'accessibilité des messages d'erreur des formulaires.

## Prochaines étapes : Drupal 8.1 et Drupal 9

Le cycle de mise à jour va s'accélérer afin d'intégrer plus rapidement de nouvelles fonctionnalités dans Drupal. Certains *patches* seront rétroportés et ajoutés à la version suivante : les problèmes mineurs seront réglés dans la version 8.1 et les changements plus importants feront partie de la version 9.

Les prochaines versions de Drupal s'adapteront aux différents navigateurs et aux attentes des utilisateurs.

---

<sup>30</sup> Toute modification importante doit passer par une série de « portes » pour assurer qu'elle soit conforme aux normes. Ces « portes » sont essentiellement des listes de contrôle qui permettent aux développeurs et aux auteurs de Patch d'évaluer leur qualité.



Les standards d'[ATAG 2.0](#) ont commencés à être intégrés dans Drupal 8, mais il reste encore du chemin à faire pour les adopter complètement.

## Potentiel d'adoption

### Bien plus qu'un CMS

Drupal permet de créer des sites complexes et personnalisés, mais pas seulement : « *Headless Drupal* » permet de déplacer l'utilisateur hors de Drupal vers des applications Javascript distinctes tels que AngularJs ou React afin de s'adapter mieux au web et à l'Internet mobile qui ne cesse de se développer.

Drupal s'adapte extrêmement bien aux appareils mobiles : les modules *Fields* (champ) et *Views* (vues) permettent de construire des blocs de contenu variés et configurer l'apparence du contenu selon l'appareil de l'utilisateur. On parle de [Responsive Design](#), et c'est exactement ce que les utilisateurs demandent puisqu'ils sont de plus en plus nombreux à accéder au web sur d'autres supports que les ordinateurs. Les usages ont également énormément évoluées au cours des dernières années et les utilisateurs s'attendent à plus d'interaction et plus de facilité.

### Un CMS accessible par défaut

Les utilisateurs doivent pouvoir prendre connaissance des contenus du Web, mais aussi et surtout accéder à leur fonctionnalité. Les personnes en situation de handicap ne doivent plus se heurter à des barrières technologiques, que ce soit en administration de systèmes, en développement ou en gestion de contenu.

Drupal constitue un réel avantage pour construire un site Web accessible puisqu'il intègre l'accessibilité dans son cœur, notamment avec des outils comme *CKEditor*, *jQuery UI* et *Chosen* qui permettent maintenant de naviguer au sein d'un site avec un clavier et sont compatibles avec les lecteurs d'écrans.

Plus de 400 développeurs ont contribué à rendre Drupal 7 accessible, et bien plus travaillent désormais sur l'accessibilité de Drupal 8.

## Pour aller plus loin

On pourra consulter les contributions de Mike GIFFORD sur les sites suivants :

<http://openconcept.ca/blog/mgifford/drupal-accessibility-example>

<http://openconcept.ca/blog/mgifford/badcamp-2014-accessibility-drupal-8>

<http://slides.com/mgifford/eight-s-accessibility-avantage#/>

## Chapitre 3 : L'amélioration de l'accessibilité d'un CMS

### L'accessibilité via les thèmes enfants dans WordPress



**Gaël POUPARD, Intégrateur web – Kosmos**

En cinq ans passés au sein d'une agence web produisant majoritairement des sites WordPress pour des TPE/PME, les pratiques de Gaël, liées à l'industrialisation des développements, l'ont conduit à capitaliser au maximum chaque investissement technique afin de les réutiliser et de les améliorer continuellement. C'est ainsi qu'il a exploré la mécanique des thèmes enfants dans WordPress, et que j'ai découvert l'accessibilité numérique.

Pour commencer, rappelons qu'un thème enfant WordPress est un **thème** qui hérite des fonctionnalités d'un autre thème, appelé thème parent. Le thème enfant est la méthode recommandée pour modifier un thème existant<sup>31</sup>.

Cependant, dans WordPress tout ne va pas dans le thème. La notion de territoire permet de déterminer le meilleur endroit pour ajouter une fonctionnalité<sup>32</sup>. Afin de distinguer le territoire des thèmes de celui des extensions, voici un algorithme basique :

- **Question 1** : La fonction doit-elle persister si vous changez de thème ?
  - Si **non** : Cette fonction est à placer dans le thème.
  - Si **oui**,
- **Question 2** : Cette fonction peut-elle être désactivée sans gêner le site au niveau fonctionnel ou visuel ?
  - Si **non** : Créez un *MU plug-in*<sup>33</sup> pour cette fonction.
  - Si **oui** : Créez un *plug-in* pour cette fonction.

Dans les thèmes WordPress, un mécanisme supplémentaire nous intéresse particulièrement : les thèmes enfants.

### Les avantages des thèmes enfants

Le principal avantage qui rend cette mécanique pertinente est de capitaliser très facilement sur son travail, y compris spécifique. Cela permet de réduire progressivement les coûts d'implémentation de telle ou telle fonctionnalité, mais aussi les coûts d'obtention de qualité (qui peuvent être très lourds quand on aborde l'accessibilité).

Avec ce mécanisme, plus vous avancerez avec l'accessibilité en tête et moins il vous coûtera de produire des sites accessibles.

<sup>31</sup> [https://codex.wordpress.org/fr:Th%C3%A8mes\\_Enfant](https://codex.wordpress.org/fr:Th%C3%A8mes_Enfant)

<sup>32</sup> <https://make.wordpress.org/themes/handbook/guidelines/plugin-territory/>

<sup>33</sup> Un *MU plug-in* ou *Must-Use plug-in*, est un *plug-in* à utiliser avant tous les autres. Bien qu'ils apparaissent dans la liste des *plug-ins* installés, un *MU plug-in* ne peut pas être désactivé, sauf en supprimant le fichier dans le répertoire « *Must-Use* »

## Organisation logistique

Les thèmes enfants sont chargés avant les thèmes parents<sup>34</sup>, il faut donc prévoir dans le thème parent la capacité d'être « déchargé » si le thème enfant le demande.

Si une fonction est appelée dans le thème parent, elle doit d'abord vérifier si le thème enfant ne la déclare pas déjà : si oui, le thème parent n'exécute pas la fonction.

Cela oblige à intégrer certaines bonnes pratiques dès la conception du thème parent, et surtout à les conserver au fil des projets. C'est une question d'hygiène technique.

On peut également évoquer :

- l'application facilitée de convention d'écriture du code ;
- la maintenabilité améliorée grâce à l'environnement qui nous est familier ;
- l'organisation et la documentation qui ne peuvent que s'améliorer au fil du temps.

## Centre de recherche

Une bonne gestion de son thème parent permet de ne pas réinventer la roue à chaque projet. C'est une base saine, ce qui permet de se focaliser sur l'amélioration *via* le thème enfant.

On peut alors se forger un thème parent accessible et ainsi économiser et capitaliser sur le temps investi pour l'accessibilité.

Il devient également facile de contre-pilier un projet pour enrichir nos outils, puisque les recherches effectuées dans le cadre d'un projet sont déjà interopérables avec nos outils.

Cela encourage la recherche et l'approfondissement des sujets, en explorant des domaines parallèles au notre.

## Les extensions

WordPress fonctionne énormément avec des extensions. La majorité des travailleurs utilisant WordPress disposent de leur flotte d'extensions favorites, utilisées systématiquement ou presque.

C'est un autre angle d'approche du même sujet : en développant ses propres extensions ou en exploitant au maximum celles dont on a l'habitude, des progrès considérables peuvent être faits<sup>35</sup>.

NB : les « *MU plug-ins* »<sup>36</sup> sont une fonctionnalité appréciable pour aider à la contribution<sup>37</sup>, ou sécuriser une installation.

---

<sup>34</sup> [http://www.rarst.net/images/wordpress\\_core\\_load.png](http://www.rarst.net/images/wordpress_core_load.png)

<sup>35</sup> Sans oublier de réinjecter ces progrès dans la communauté à plus ou moins long terme

<sup>36</sup> [http://codex.wordpress.org/Must\\_Use\\_Plugins](http://codex.wordpress.org/Must_Use_Plugins)

<sup>37</sup> J'en propose quelques-uns (qui ne sont pas à l'état de l'art, soyez prudents) : <https://github.com/ffoodd/ffeeeedd--extensions/tree/master/mu-plugins>

Ils sont particulièrement intéressants puisqu'ils ne peuvent pas être désactivés *via* l'interface d'administration.

## Les équivalents sur d'autres CMS

*A priori*, seul Drupal possède un mécanisme équivalent et bien plus évolué<sup>38</sup> (à mon sens).

Leur approche est extrêmement intéressante puisque chaque thème peut adopter plusieurs sous-thèmes. On peut ainsi imaginer des sous-thèmes spécialisés, ou une gestion par couche de l'interface.

La majorité des autres CMS conseille de modifier directement les thèmes... Cela ne permet malheureusement pas de faciliter la maintenance, les évolutions et le recyclage. Le travail est difficilement capitalisable.

---

<sup>38</sup> <https://www.drupal.org/node/225125>

## L'accessibilité dans le monde Joomla !



### Ariane ANDURAND, Responsable du Pôle Web - C3RB Informatique

Ariane travaille dans le Web depuis les années 2000. Experte AccessiWeb en évaluation depuis 2011, la sensibilisation de ses collègues développeurs, formateurs et contributeurs au domaine de l'accessibilité est devenue sa priorité. Membre du Groupe de Travail AccessiWeb, elle travaille exclusivement sur le CMS Joomla! depuis 2008. Elle est experte de l'intégration de ce CMS et de certains composants qui gravitent autour.

## Prise en compte de l'accessibilité dans Joomla! : un bref historique

Dans un premier temps, les initiatives pour apporter de l'accessibilité à Joomla! étaient individuelles.

La conceptrice allemande des *templates* « Beez », Angie RADTKE<sup>39</sup>, a proposé un *template front* accessible pour la version Joomla!1.5, réédité en Joomla!2.5<sup>40</sup>.

Aujourd'hui avec la version 3 de Joomla! les *templates* « Beez » ont disparu du pack par défaut. Deux initiatives individuelles ont été recensées, celle de [C3RB Informatique](#) et celle de [Francesco ZANIOL](#). Ces initiatives ont donné naissance à deux *templates* : Zhong et RGAAC3rb.

La feuille de route de la *Core Team* de Joomla! prévoit en Joomla! 3.10 la création et l'intégration d'un nouveau *template* accessible pour l'interface d'administration afin de remplacer les deux *templates* existants. La prise en compte de l'accessibilité se fera donc également côté administration ce qui est essentiel pour un CMS. L'objectif est de proposer un design unique, robuste et convivial.

Le système de *bug tracker* sur GitHub mis en place par les *développeurs* de Joomla! permet également de faire remonter tout problème d'accessibilité du CMS<sup>41</sup>

## Deux exemples de *templates* accessibles en Joomla!3

### Zhong template

Francesco ZANIOL a développé son *template* Zhong au cours de sa thèse à l'université.

Tous les *templates* proposés par Francesco sont basés sur le Framework Zhong ayant pour but de proposer une base solide d'accessibilité pour la construction de modèles<sup>42</sup>. Ce *framework* a pour ambition de combiner d'un côté élégance du design, bonnes pratiques et accessibilité pour le développement *front-end*.

<sup>39</sup> <https://twitter.com/angieradtke>

<sup>40</sup> lire l'article sur le site Cooate: <http://cocoate.com/fr/j25fr/le-template-beez>

<sup>41</sup> voir un exemple de remontées: <http://issues.joomla.org/tracker/joomla-cms/6467>

<sup>42</sup> <http://www.accessibletemplate.com/zhong/about-zhong>

Les composantes d'accessibilité sont basées sur la norme [WCAG 2.0](#)<sup>43</sup>. Le code basé sur l'utilisation d'éléments compressés (CSS, HTML et minimisation du JavaScript) est performant. Il inclut un code PHP modulaire et l'utilisation de SASS pour dissocier le style du cœur et celui du thème.

Le Zhong Framework permet également de modifier profondément les couleurs et les styles de chaque élément tout en proposant des plus comme des icônes sociales, des boutons, des fonctions de [Responsive Design](#), le choix de polices Google importées, le paramétrage possible de son code Google Analytics, la compatibilité avec Bootstrap. Le *template* dispose d'un guide complet d'installation.

Pour le moment le Zhong Framework est compatible avec Joomla 2.5 and Joomla 3.x, cependant il est prévu de porter la compatibilité de celui-ci vers d'autres plateformes comme Drupal et Wordpress.

Une version gratuite du *template* est disponible en [téléchargement libre](#)<sup>44</sup>

### **RG AAC3rb Template**

C3rb Informatique a développé un *template* Joomla!3.X conforme aux normes d'accessibilité en vigueur. Ce *template* a été nommé RG AAC3rb. Conscient de la force de la communauté Joomla! et pour favoriser la réflexion de la communauté d'utilisateurs, C3rb Informatique a opté pour une mise à disposition libre<sup>45</sup>. Sa présentation a eu lieu aux Joomla!Day 2015 de Nice.

RG AAC3rb *Template* intègre des balises pour l'implémentation ARIA et certaines vues du [core](#) de Joomla! sont mises à disposition dans le *template* pour améliorer l'accessibilité du code d'origine. Les équipes de C3rb soumettent des issues au fur et à mesure à la *Core Team* sur le GitHub de Joomla!. Un [plug-in](#) a été développé pour gérer les conflits liés à l'intégration de Bootstrap 3 (Plug-in `plg_system_rgaac3rb`). Le choix s'est porté sur Bootstrap car il prend en compte les développements du *Paypal Accessible Plug-in* par exemple.

RG AAC3rb est fourni avec des dépendances à jour et maintenable via GRUNT, Javascript Task Runner<sup>46</sup>.

---

<sup>43</sup> Lire notamment : <http://www.accessibletemplate.com/zhong/features>

<sup>44</sup> <http://www.accessibletemplate.com/zhong/free-download>

<sup>45</sup> url : <http://c3rb.org/jd15fr>

<sup>46</sup> <http://gruntjs.com/>

## Chapitre 4 : Etude comparative de la capacité des CMS à produire des contenus accessibles



### Olivier NOURRY, Consultant et Formateur Accessibilité - Smile

Professionnel du Web depuis 2000, consultant et formateur en accessibilité du Web depuis 2006. Riche d'une expérience en développement, en gestion de projet et en assistance MOE et MOA, Olivier accompagne les organisations dans la mise en œuvre de l'accessibilité au sein de leurs projets de création ou refonte de sites Web. Il forme développeurs, responsables de projet et rédacteurs à l'application des recommandations du RGAA et des bonnes pratiques.

### Claire BIZINGRE, Consultante formatrice Web - Accesbilis

Claire intervient dans les projets Web pour assurer la prise en compte de l'accessibilité à chaque étape du projet, et pour la rédaction d'attestations de conformité au RGAA. Elle réalise des audits et propose des corrections pour améliorer l'accessibilité des sites Web. Elle anime des formations en accessibilité mais aussi en HTML/CSS, en conception d'interface Web et en WordPress. Elle est formatrice au Centre National de la Fonction Publique Territoriale (CNFPT) et enseignante vacataire à l'Université Paris Est de Créteil (UPEC).



### Edouard CUNIBIL, Drupangéliste - Happyculture ScopARL

Développeur Drupal depuis 2009, Edouard a accompagné des entreprises locales dans le développement de leurs projets de toutes sortes, de la simple vitrine à l'intranet social en passant par des projets e-commerce. Parallèlement, depuis 2010, Il s'investit pleinement dans la communauté Drupal en tâchant de dédier une partie de son temps à la contribution sous toutes ses formes : maintenance de modules, modifications du cœur, organisation de rencontres mensuelles et d'événements d'envergure nationale (DrupalCamps) et internationale (Drupal Developer Days).



### Jacques PYRAT, Expert SPIP et formateur — Pyrat.net

Professionnel du Web depuis 1997, Jacques découvre SPIP en 2002. Immédiatement séduit par son respect des règles typographiques, il adapte SPIP pour tous ses développements. Jacques est fortement engagé dans la communauté SPIP (pseudo RealET) et contribue activement à son écosystème. Le squelette SoyezCréateurs développé depuis 2003 se veut un modèle d'accessibilité. Ses clients bénéficient largement de cette accessibilité pour la qualité de leur référencement.



### Mickael RUZAFKA, Intégrateur - C3rb informatique

Professionnel du Web depuis 2005, Intégrateur, Web designer, Mickael réalise des sites internet, portails documentaires pour les médiathèques, bibliothèques et bibliothèques départementales de France. Son travail consiste aussi en la création et l'évolution d'un "kit de démarrage" pour la réalisation de multiples sites avec le CMS Joomla! : [https://github.com/c3rb-org/template\\_RGAA\\_C3rb](https://github.com/c3rb-org/template_RGAA_C3rb). Il participe à la recherche et à l'évolution des produits de la société C3rb informatique.



## Introduction : Contexte et objectifs

Le choix du CMS approprié est une des clés de réussite - ou pierre d'achoppement - d'un projet accessibilité. La question a été souvent posée à BrailleNet : Quel est le meilleur CMS à utiliser ?

Or à cette question, il n'y a pas de réponse simple et unique. Que l'on trouve cela fâcheux ou rassurant, la réponse dépend de nombreuses fonctionnalités du CMS et surtout de la façon dont ces fonctionnalités sont mises en œuvre dans le contexte d'un projet précis.

C'est pourquoi nous avons décidé avec Olivier NOURRY de mener une étude comparative de plusieurs CMS très couramment utilisés, en faisant appel à quatre développeurs experts de ces CMS.

Si l'on se réfère au standard [ATAG 2.0](#)<sup>47</sup> du W3C, l'accessibilité des CMS, s'évalue à 3 niveaux :

- Accessibilité des interfaces
- Accompagnement de l'utilisateur dans la création de contenus accessibles
- Accessibilité des contenus mis en ligne

L'étude se concentre sur le 3<sup>ème</sup> point ; elle vise à évaluer la capacité d'un CMS donné à produire des contenus qui respectent les recommandations d'accessibilité [WCAG 2.0](#).

Dans cette première version elle prend en compte, quatre CMS, parmi les plus utilisés<sup>48</sup> : Drupal, Joomla!, SPIP et WordPress. Les experts ayant contribué à l'étude étant

- Pour Drupal : Edouard CUNIBIL, Happyculture ScopARL
- Pour Joomla! : Mickael RUZAFI, C3rb informatique
- Pour SPIP : Jacques PYRAT, Pyrat.net
- Pour WordPress : Claire BIZINGRE, Accesbilis

L'étude fournit des éléments concrets de réponses à des exigences d'accessibilité qui entrent dans le champ soit des [contributeurs](#), soit des [développeurs](#) et confirme que les choix de paramétrage, de saisie, ou de modification du CMS sont déterminants dans l'accessibilité des contenus produits.

## Méthodologie

### Raisonnement

Rendre des contenus contribués via un CMS accessibles requiert de croiser deux compétences clés :

- Compréhension des exigences d'accessibilité
- Maîtrise du CMS

Le postulat de départ est que la compétence CMS est plus longue et difficile à acquérir que celle sur l'accessibilité. Pour initier l'étude, quatre spécialistes des quatre CMS retenus pour l'étude, ont été sollicités et ont accepté de soumettre un CMS à une grille d'évaluation que nous leur avons proposée.

<sup>47</sup> <http://www.w3.org/TR/ATAG20/>

<sup>48</sup> D'après [w3techs](#) les parts de marchés de WordPress, Joomla! et Drupal au plan mondial sont respectivement 24,3%, 2,8% et 2,%. Quant à SPIP il a de nombreux utilisateurs en France, notamment dans le secteur public.



## Grille d'évaluation

La grille d'évaluation a été établie sur la base du [référentiel RGAA 3.0](#)<sup>49</sup>. Elle consiste en une liste de besoins à satisfaire. Chaque besoin a été défini en fonction des critères suivants :

- La satisfaction d'un besoin permet de satisfaire un ou plusieurs critères A ou AA
- Les fonctionnalités offertes par le CMS ont un impact direct sur la capacité du contributeur à satisfaire le besoin (au-delà de sa fonction première qui est de générer du code HTML),

Sur cette base, des critères tels que « les liens doivent être explicites », ou « une transcription textuelle des contenus multimédia doit être disponible », n'ont pas été inclus dans la grille; car, dans les deux cas, la solution technique consiste à produire du code HTML, ce qui est la fonction de base d'un CMS. De tels critères ne permettent donc pas de différencier les CMS entre eux quant à l'accessibilité des contenus produits.

De même, les critères de respect des contrastes de couleurs, par exemple, ne sont pas pris en compte : cet aspect relève soit de la charte graphique (thème), soit des choix de couleurs faits par l'utilisateur. La capacité du CMS n'est pas en cause.

La grille complète des critères pris en compte est fournie à la fin de cet article.

## Utilisation

Pour chaque besoin de la grille d'évaluation, il a été demandé à l'expert du CMS de proposer une solution technique ou rédactionnelle.

Chaque expert avait toute latitude pour la méthode sélectionnée ; la seule instruction était de proposer une méthode réaliste, cohérente avec le CMS, et que lui-même (ou elle-même) auraient adoptée en situation de projet.

Cette approche permettait d'éviter des manipulations qui seraient disqualifiées en situation réelle et de mieux identifier les carences des CMS en version de base, ou de leurs modules les plus populaires.

Les spécialistes consultés ont rempli chacun une feuille d'un tableau Excel, nommée avec le nom du CMS concerné. Pour chaque besoin, ils ont rempli les colonnes suivantes :

[1] Résultat :

- OK : le besoin est satisfait ;
- KO : le besoin n'est pas satisfait ;
- NA : la demande n'est pas applicable (par exemple, le CMS ne dispose pas des fonctionnalités correspondantes) ;
- Limité : une partie du besoin est satisfaite, ou avec restrictions.

[2] Méthode

- *Core* : le besoin est satisfait via une fonctionnalité du noyau du CMS ;
- Module : le besoin est satisfait via un module additionnel (*plug-in*) du CMS ;

---

<sup>49</sup> <http://references.modernisation.gouv.fr/referentiel-technique-0>

- **RTE** : le besoin est satisfait via l'éditeur de Texte Riche (*Rich Text Editor*) inclus par défaut au CMS ;
- **Thème** : le besoin est satisfait via une fonctionnalité du thème (*skin*, masque, etc.) du CMS ;
- Saisie manuelle : le besoin est satisfait via une opération de saisie manuelle, dans l'interface de contribution ou le RTE du CMS.

[3] Phase :

- Conception : relève plutôt du développeur (installation ou paramétrage de module, modification ou paramétrage de thème, etc.) ;
- Utilisation : relève plutôt du contributeur (saisie dans le RTE ou l'interface, etc.).

[4] *Score*, Max :

- La valeur Max permet de pondérer le besoin par l'impact utilisateur (un besoin ayant une note élevée correspond à un fort impact utilisateur) ;
- Le *score* peut être 0 (le besoin n'est pas satisfait), Max (le besoin est satisfait), ou une valeur intermédiaire (le besoin est satisfait partiellement, ou via une manipulation complexe). Le spécialiste apprécie ce *score* en fonction de son expérience.

[5] Remarques : permet de préciser certains points de la manipulation.

## Cas particuliers

Pour certains aspects des exigences d'accessibilité, il n'a pas été possible de définir un besoin suffisamment précis.

Ainsi, la [thématique scripts](#)<sup>50</sup> se réfère à une grande variété de situations, qu'il est difficile de ramener à quelques tests génériques. De plus, cet aspect est fortement impacté par les modules ou les thèmes installés. C'est pourquoi la vérification se fait à la fois sur les widgets disponibles dans la version de base du CMS et sur la possibilité de remplacer la bibliothèque JavaScript existante par celle de son choix.

De même pour le critère de [conformité du code](#)<sup>51</sup> : on vérifie ici que le code du CMS dans sa version de base, avec les modules et le thème par défaut, est conforme ; et qu'il est possible de remplacer l'éditeur de Texte Riche, pour faire le choix d'un outil plus performant.

## Résultats

En général, il ressort de l'étude que tous les CMS étudiés obtiennent de très bons résultats, moyennant quelques adaptations ou manipulations. Aucun n'obtient la note maximale cependant ; les critères d'accessibilité correspondants à des évolutions récentes des WCAG ou RGAA, tenant compte de HTML5 et ARIA, notamment, sont encore rarement implémentés.

L'étude a été l'occasion, pour les quatre spécialistes sollicités, d'identifier des carences dans la satisfaction des critères d'accessibilité du RGAA 3, dans leur CMS de prédilection. Ils ont pu remonter ces anomalies auprès des équipes de développement des CMS concernés.

---

<sup>50</sup> <http://references.modernisation.gouv.fr/referentiel-technique-0#title-127-scripts>

<sup>51</sup> <http://references.modernisation.gouv.fr/referentiel-technique-0#title-critre-82-a-pour-chaque-page-web-le-code-source-est-il-valide-selon-le-type-de-document-spcifi-hors-cas-particuliers->

Le fichier Excel des résultats détaillés, CMS par CMS, peut être téléchargé à l'adresse suivante : [http://www.accessiweb.org/tl\\_files/doc\\_telechargement/gta21/resultats-de-l-etude-CMS.xlsx](http://www.accessiweb.org/tl_files/doc_telechargement/gta21/resultats-de-l-etude-CMS.xlsx).

## Conclusions

La méthode proposée pour comparer de la capacité des CMS à produire des contenus accessible nous semble avoir fourni des résultats utiles et reproductibles.

C'est pourquoi BrailleNet envisage de faire appel à des experts pour des contributions similaires pour d'autres CMS, ou d'autres versions de ces CMS. Il suffira d'ajouter au tableau récapitulatif autant d'onglets que nécessaire, afin de mettre à disposition en un lieu unique tous les résultats.

Par ailleurs, une étude similaire, faisant appel également à des spécialistes de CMS, pourrait être réalisée pour l'accessibilité des interfaces, avec pour objectif d'identifier les carences, et de proposer des améliorations.

## Grille d'analyse utilisée pour l'étude

Thématique	Critère AccessiWeb / RGAA3.0	Besoin
Images	1.1	Insérer une image décorative (<img>, alt vide, title absent)
Images	1.2	Insérer une image vectorielle décorative (<svg>, role="img", pas de label aria, pas d'attribut title, <desc> et <title> absents ou vides)
Images	1.3	Insérer une image informative (<img>, alt non vide, title absent ou identique à alt)
Images	1.4	Insérer une image vectorielle informative (<svg>, role="img", aria-label ou <desc>, identique à attribut title si présent)
Images	1.5	Désactiver les CAPTCHAs visuels
Images	1.6	Activer les CAPTCHAs visuels, avec une alternative textuelle modifiable (ou appropriée), et possibilité de définir un autre type de CAPTCHA
Images	1.7	Insérer une image (<img>) avec légende: - image et légende dans une balise <figure role="group"> - alt paramétrable - si title présent, identique à alt
Images	1.8	Insérer une image vectorielle (<svg>) avec légende: - image et légende dans une balise <figure role="group"> - aria-label ou <desc> paramétrable - si title présent, identique à aria-label ou <desc> utilisés
Multimédia	4.2	Associer une audiodescription à un contenu vidéo
Multimédia	4.3	Associer des sous-titres à un contenu média (audio ou vidéo)
Multimédia	4.5	Insérer des contenus média (audio ou vidéo) sans autoplay
Multimédia	4.6	Lecteur média utilisable au clavier
Multimédia	4.7	Lecteur média disposant des fonctions lecture, pause, stop, réglage du volume, activation sous-titres, activation audiodescription
Multimédia	4.8	Lecteur média compatible avec les lecteurs d'écran (restitution des rôles, valeurs, changement d'état des éléments d'interface)
Tableaux	5.1	Les tableaux de mise en forme insérés par le CMS ont les caractéristiques suivantes: - role="presentation" - compréhensibles en lecture linéaire - pas d'éléments caption, th, thead, tfoot - pas d'attributs scope, headers, axis, colgroup
	5.2	Les tableaux de données simples ont les caractéristiques suivantes: - élément caption, contribuable - th pour les en-têtes, avec attribut scope pertinent

Thématique	Critère AccessiWeb / RGAA3.0	Besoin
	5.3	Les tableaux de données complexes ont les caractéristiques suivantes: - élément caption, contribuable - th pour les en-têtes, avec couples id/headers pour créer les associations en-têtes/cellules de données
Liens	6.1	Définir des titres de liens
Scripts	7.1	<i>Quelle est la bibliothèque JavaScript utilisée par défaut?</i>
Scripts	7.2	Choisir sa propre bibliothèque JavaScript
Scripts	7.3	Les widgets de la configuration par défaut sont compatibles avec les technologies d'assistance
Scripts	7.4	Les widgets de la configuration par défaut sont utilisables au clavier et à la souris
Éléments obligatoires	8.1	Le code généré est conforme (configuration par défaut)
<i>Éléments obligatoires</i>	8.2	<i>Quel est l'éditeur texte riche utilisé par défaut?</i>
Éléments obligatoires	8.3	Choisir son propre RTE
Éléments obligatoires	8.4	Définir la langue par défaut de la page
Éléments obligatoires	8.5	Gérer les attributs lang sur les textes contribués
Éléments obligatoires	8.6	Gérer les attributs dir sur les textes contribués
Éléments obligatoires	8.7	Définir les titres de page
Éléments obligatoires	8.8	Dans les titres de page générés automatiquement, les balises HTML sont supprimées
Éléments obligatoires	8.9	Dans les contenus paginés (résultats de recherche, listes d'articles, etc.), le numéro de page figure dans le titre de page
Structuration	9.1	Définir des titres de sections pertinents
Structuration	9.2	Dans le thème par défaut, structurer de façon pertinente les pages avec les éléments suivants: - balise <header> - balise <nav>, réservée aux navigations principale et secondaire - balise <main> unique - balise <footer>
Structuration	9.3	Définir des listes <ul>, <ol>, <dl>
Structuration	9.4	Définir des citations <q> et <blockquote>
Formulaires	11.1	Dans les formulaires générés automatiquement, tous les champs ont une étiquette pertinente
Formulaires	11.2	Dans les formulaires contribués, définir une étiquette pour tous les champs
Formulaires	11.3	Dans les formulaires générés automatiquement, les champs de même nature sont regroupés avec la balise <fieldset>, et ont une balise <legend> pertinente
Formulaires	11.4	Dans les formulaires contribués, regrouper les champs de même nature avec la balise <fieldset>, et définir une balise <legend> pertinente
Formulaires	11.5	Dans les formulaires générés automatiquement, les listes de choix <select> sont structurées avec une balise <optgroup> si nécessaire
Formulaires	11.6	Dans les formulaires contribués, structurer les listes de choix <select> avec une balise <optgroup>
Formulaires	11.7	Dans les formulaires générés automatiquement, tous les boutons ont un intitulé pertinent
Formulaires	11.8	Dans les formulaires contribués, définir un intitulé pour tous les boutons
Formulaires	11.9	Dans les formulaires, tous les champs obligatoires sont signalés de manière accessible

Thématique	Critère AccessiWeb / RGAA3.0	Besoin
Formulaires	11.10	Dans les formulaires, pour tous les champs obligatoires, le type de données attendu est indiqué de manière accessible
Formulaires	11.11	Dans les formulaires, pour chaque erreur de saisie, le type de données attendu est indiqué de manière accessible
Navigation	12.1	Pouvoir insérer au moins 2 éléments parmi ces 3: - plan du site - menu de navigation - moteur de recherche
Navigation	12.2	Dans les collections de pages, insérer des liens de navigation (pagination) dans la collection
Navigation	12.3	Dans le thème par défaut, identifier les zones de page avec les éléments suivants: - attributs id, ou ancres nommées, sur les groupes de liens importants - attributs id, ou ancres nommées, sur la zone de contenu principal - rôles ARIA banner, main, contentinfo, search (uniques dans la page), et navigation
Navigation	12.4	Dans le thème par défaut, présence et paramétrage de liens d'évitement
Navigation	12.5	Quel que soit la configuration de la page, l'ordre de tabulation est cohérent
Navigation	12.6	Pas de piège au clavier avec les widgets de la configuration par défaut

## Glossaire

<b>AccessiWeb HTML5/ARIA</b>	Le référentiel AccessiWeb HTML5/ARIA, publié par BrailleNet, est sorti fin 2013 et permet la prise en charge de l'accessibilité d'un site Web dans un contexte technique orienté interaction utilisateur. Ce référentiel a été repris par l'administration dans son RGAA3.0.
<b>API</b>	Un <i>Application Programming Interface</i> (interface de programmation) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert d'interface par laquelle un logiciel offre des services à d'autres logiciels.
<b>ATAG</b>	Les <i>Authoring Tools Accessibility Guidelines</i> sont un standard du W3C destiné à aider les développeurs d'outils d'édition de contenus Web à faciliter la production de contenus conformes aux recommandations WCAG sur l'accessibilité des contenus web.
<b>Back-office</b>	La partie d'un site Web qui n'est pas accessible aux utilisateurs finaux, et qui permet de l'alimenter et le mettre à jour.
<b>Contributeur</b>	Personne qui crée et administre des contenus au travers de l'interface de contribution. Ses interventions se limitent à de la saisie et à, éventuellement, du paramétrage. Typiquement, le contributeur ne modifie pas le code du CMS, de ses modules ou de son thème.
<b>Commits</b>	Contribution à l'évolution d'un logiciel dans son code source.
<b>Core</b>	Ensemble de répertoires et de fichiers de configuration nécessaires au fonctionnement d'un système d'information.
<b>Développeur</b>	Personne qui intervient sur le code du CMS, de ses modules ou de son thème ; éventuellement via l'installation d'éléments additionnels, ou leur paramétrage. Ces interventions ont un caractère technique, requérant fréquemment des compétences de programmation ou d'administration du système informatique.
<b>Framework</b>	Un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel.
<b>Front-office</b>	La partie d'un site Web accessible aux utilisateurs finaux, par opposition au back-office.
<b>Patch</b>	Correctif apporté à un logiciel.
<b>Plug-in</b>	Un module qui complète un logiciel hôte pour lui apporter de nouvelles fonctionnalités. Cette extension des fonctionnalités est prévue, par opposition aux ajouts non prévus apportés à l'aide de correctifs (patches).
<b>Responsive design</b>	Un site web responsive, ou adaptatif en français, est un site web dont la conception vise, grâce à différents principes et techniques, à offrir une expérience de consultation confortable sur une large gamme d'appareils (moniteurs d'ordinateur, smartphones, tablettes, TV, etc.).
<b>RGAA 3.0</b>	<i>Référentiel Général d'Accessibilité des Administrations</i> , version 3. Référentiel officiel de l'État français, repris d'AccessiWeb HTML5/ARIA; qui répertorie thème par thème l'ensemble de critères de vérification du respect des bonnes pratiques d'accessibilité.
<b>RTE</b>	Rich Text Editor, voir WYSIWYG

<b>Sprint</b>	Un rassemblement de développeurs impliqués dans un projet qui se concentrent sur le développement de ce projet pendant une durée déterminée. Le sprint est dirigé par un coach qui suggère les tâches, en suit l'avancement et s'assure qu'aucun participant n'est bloqué.
<b>Template</b>	Selon le CMS, on peut parler de « <i>template</i> » ou de « thème », les deux se référant essentiellement à une collection de fichiers de code et d'images qui déterminent l'aspect visuel du site Web. Drupal et WordPress parlent de thème, alors que Joomla ! parle de <i>template</i> .
<b>Thème</b>	Voir <i>Template</i>
<b>UAAG</b>	Les <i>User Agent Accessibility Guidelines</i> pour l'accessibilité des navigateurs, sont un standard s'adressant aux développeurs de logiciels de navigation et de consultation de contenu Web (applications Web, lecteurs multimédia, plug-ins, technologies d'assistance, etc.).
<b>WAI</b>	L'Initiative sur l'Accessibilité du web ou Web Accessibility Initiative (WAI) fut lancée en avril 1997 par le World Wide Web Consortium (W3C). La principale mission de la WAI est de proposer des solutions techniques pour rendre le World Wide Web accessible aux personnes handicapées et d'une manière plus générale à toute personne sans nécessiter de pré-requis particulier.
<b>WAI-ARIA</b>	WAI-ARIA ( <i>Web Accessibility Initiative - Accessible Rich Internet Applications</i> ) est une spécification technique du W3C. L'objectif est d'accroître l'accessibilité des contenus dynamiques et des composantes des interfaces dynamiques développées à l'aide d'Ajax, HTML, Javascript et des technologies associées. ARIA décrit comment ajouter de la sémantique et des métadonnées aux contenus HTML afin de rendre les contrôles d'interface et les contenus dynamiques plus accessibles.
<b>W3C</b>	Le World Wide Web Consortium est un organisme de normalisation à but non lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du World Wide Web (HTML5, HTML, XHTML, XML, CSS, etc.).
<b>WCAG</b>	Les Règles pour l'accessibilité des contenus Web ( <i>Web Content Accessibility Guidelines</i> ) sont un standard du W3C proposant un large éventail de recommandations pour rendre les contenus Web plus accessibles et s'adressant à tous les distributeurs de contenu sur le Web. Après les WCAG1.0 publiées en 1999, les WCAG2.0 sont le standard officiel depuis le 11 décembre 2008.
<b>WYSIWYG</b>	Un WYSIWYG, ou <i>What you see is what you get</i> , est une interface utilisateur qui permet au producteur de contenus de composer et éditer ses contenus directement à l'écran et de voir à quoi ressemblera son document tel qu'il sera publié.

## Partenaires premium

---



[Tester Tanaguru my.tanaguru.com](https://my.tanaguru.com)

# L'outil le plus puissant pour l'accessibilité du Web

## L'accessibilité, simplement

La voie la plus simple pour optimiser votre

Aujourd'hui, il est difficile et fastidieux de vérifier l'accessibilité à chaque étape d'un projet Web. Ce travail demande des ressources qualifiées et un temps de traitement important. Tanaguru minimise les coûts liés à l'évaluation de l'accessibilité des développements. L'automatisation permet, d'une manière simple et fiable, d'apporter aux développeurs une certaine autonomie.

## Tanaguru automatise 173 tests d'accessibilité du RGAA 3

L'évaluation d'une page, d'un site entier ou d'une application web est fiable, intuitive et elle-même accessible.



### Fiable

Les résultats de Tanaguru sont fiables.

Tanaguru vous fournit tous les indicateurs nécessaires à votre prise de décision.



### Automatisé

Repousser les limites de l'automatisation est notre motivation.

Gagner en productivité fait partie de l'ADN de Tanaguru.



### Facile

L'accessibilité peut être complexe.

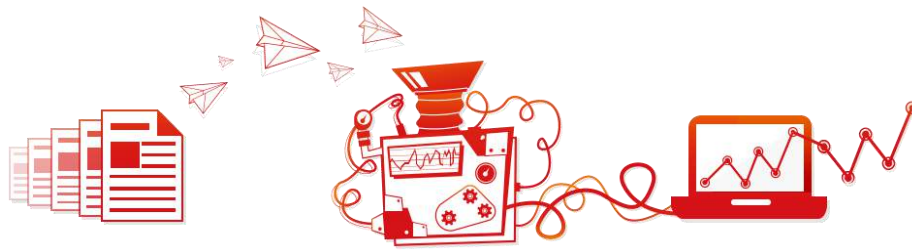
Tanaguru apporte les solutions pour initier ou contrôler la conformité de vos sites web.

Contact : Manuel Pereira

Responsable commercial accessibilité numérique  
[mpereira@oceaniconsulting.com](mailto:mpereira@oceaniconsulting.com)



**01.80.48.38.15**



### Audit de page

Tanaguru évalue instantanément une page ou un ensemble de pages web...



### Audit de site

Grâce à son crawler, Tanaguru parcourt la totalité de votre site internet...



### Audit de fichier

Auditez vos gabarits avant même leur mise en production...



### Audit de scénario

Validez les applications Javascript, les intranets ou le parcours d'achat d'un site e-commerce...



### Jenkins

Réaliser ses tests (unitaires, fonctionnels, non-régression...) dans une chaîne d'intégration continue est un atout majeur pour la qualité de ses productions. Afin d'ajouter les tests d'accessibilité, Tanaguru s'intègre dans cette chaîne grâce au **Tanaguru Jenkins Plugin**.

## Tanaguru, c'est aussi un écosystème de logiciels libres dédiés à l'accessibilité

**KBAccess.org** est une base collaborative de bons et mauvais exemples d'accessibilité.

**Contrast-Finder** mesure les contrastes pour l'accessibilité mais surtout propose des résultats valides.

### Offre Océane Consulting

- Tanaguru SAAS un service performant à un coup compétitif, toujours à jour.
- Tanaguru Server, profitez de votre propre instance de Tanaguru.
- Maintenance et support, l'assurance d'un outil toujours disponible.
- Développement, s'adapter à vos besoins spécifiques.
- Formation à l'utilisation de Tanaguru, comprendre les fonctions avancées de l'outil.





## Rendons la digitalisation accessible

V-Technologies s'est intéressé à l'accessibilité dès 2004, alors que la loi sur le handicap n'était pas encore promulguée, et encore moins le décret concernant l'accessibilité numérique. La mise en place par BrailleNet des premières formations dans le domaine a été un vrai déclencheur. Au fil des années, l'accessibilité est demeurée en filigrane de nos réalisations, sans que soit exprimée de véritable demande chez nos clients, qu'ils soient privés ou même publics. Pour ces derniers, l'exigence d'un respect du Référentiel Général d'Accessibilité pour les Administrations (RGAA) était un paragraphe obligatoire, à placer dans leur Cahier des Clauses Techniques Particulières (CCTP).

La chance de participer depuis l'an dernier, aux côtés de BrailleNet et d'Access42, au groupement en charge de "l'amélioration de l'accessibilité des systèmes d'information des employeurs publics", autour du RGAA 3.0, nous permet désormais de contribuer plus largement à cette amélioration. En effet, il ne s'agit plus seulement d'accompagner des organismes dans leur démarche ou de produire des applications et contenus plus accessibles, mais également de fournir à la communauté des développeurs des ressources directement exploitables, leur faisant ainsi gagner un temps précieux dans leurs travaux de "digitalisation accessible".

## Des experts et une équipe formés

Le premier salarié de V-Technologies, devenu en 2004 expert en évaluation de l'accessibilité (EAE), a immédiatement transmis le "virus" aux équipes. C'est ainsi que, de 2005 à 2014, 6 autres EAE (chefs de projet, développeurs, intégrateurs) ont suivi la formation AccessiWeb. Leurs connaissances ont été mises à jour début 2015, à l'occasion de l'arrivée du RGAA 3. Cette année a vu également la montée en compétence de 2 nouveaux chefs de projet, d'une chargée de communication et d'un directeur de production, formés à l'évaluation, mais aussi de 6 développeurs ou intégrateurs qui ont acquis des compétences, techniques... pour la mise en œuvre. Dans l'équipe de production, une personne sur deux est désormais formée à l'accessibilité. Ils constituent chez V-Technologies notre mini GTA (Groupe de Travail AccessiWeb) qui se réunit régulièrement et partage expériences et savoir-faire.

Ce sont les EAE AccessiWeb qui assurent les audits pour les sites de nos clients (voir ci-après l'exemple du Bas-Rhin) mais aussi, dans le cadre du groupement évoqué plus haut, une partie des inspections pour le tout jeune label "e-accessible" de l'État. Ce fut le cas pour la labellisation du site du Conseil départemental du Pas de Calais, premier à obtenir ce label, avec un niveau "5 étoiles".

## Des applications riches accessibles

Les recommandations RGAA 3 et le label "e-accessible" ne sont pas réservés aux sites Internet. Ils peuvent s'appliquer à des intranets, des extranets et des applications Web avec des fonctionnalités avancées. Les navigateurs Web ont considérablement évolué ces dernières années. Ils permettent désormais de proposer des applications complexes. Il est maintenant possible de créer en ligne un client e-mail, ou encore une suite bureautique complète.

Nous réalisons une veille constante sur les outils permettant de rendre les applications de plus en plus riches. Toutefois, ces outils ne fournissent par défaut aucun système de gestion au clavier ou de support des technologies d'assistance. Notre équipe est ainsi formée à la création d'applications riches accessibles avec les technologies récentes. Nous travaillons en particulier avec les bibliothèques et frameworks AngularJs, React, BackbonesJs. Nous créons et migrons des applications Web en suivant les grilles de test ARIA (Accessible Rich Internet Applications) du référentiel RGAA 3.

Pour partager avec le plus grand nombre nos recherches sur l'accessibilité des bibliothèques et frameworks, nous sommes également contributeurs de projets open source. Ceci nous permet de mettre à disposition de la communauté des corrections et des créations de "design patterns" accessibles, comme :

- [https://github.com/DISIC/rgaa\\_bibliotheques\\_javascript](https://github.com/DISIC/rgaa_bibliotheques_javascript) (Modules javascripts d'accessibilité des bibliothèques JQueryUI, AngularUI Bootstrap et React Bootstrap) - réalisé pour le compte du SGMAP, dans le cadre du groupement en charge de l'amélioration de l'accessibilité (voir plus haut), en collaboration avec Access42 (pour les grilles) et Brailletnet (pour les tests de lecteurs d'écran) ;
- <https://github.com/felixgirault/rea11y> (Création de patterns ARIA en React)
- <https://github.com/angular-ui/bootstrap> (Correction de patterns ARIA avec AngularJs)

D'autres chantiers autour des CMS (Content Management System) et des applications mobiles sont en cours.

## De l'accompagnement au long cours : l'exemple du Conseil départemental du Bas-Rhin

V-Technologies accompagne, depuis le début de cette année, le Conseil départemental du Bas-Rhin dans la mise en accessibilité de son site principal, bas-rhin.fr, un site relativement ancien et développé en environnement Microsoft Windows (SharePoint). Les enjeux pour cette collectivité sont principalement l'amélioration de l'accessibilité de ce site Web et de tous ses contenus (images, vidéos, articles, documents..) mais aussi, à plus long terme, l'adoption d'une démarche d'accessibilité pour l'ensemble de ses projets de réalisation numérique.

La première étape de ce processus a été pour nous la réalisation d'un audit du site existant. Il a permis d'identifier un certain nombre d'anomalies représentatives des points de non-conformité et a mis en lumière des besoins de préconisations. Tous les points de non-conformité ou axes d'amélioration ont fait l'objet de préconisations unitaires avec une priorisation en fonction de l'impact sur les utilisateurs (nature du problème, récurrence) et de la facilité de mise en œuvre (coût technique et humain), de manière à élaborer un plan de mise en conformité réaliste.

En amont des corrections à apporter, une formation de l'ensemble des intervenants - deuxième étape du processus - a été effectuée pour une bonne sensibilisation aux enjeux et aux bonnes pratiques en matière d'accessibilité. Pour l'équipe de la DSI (Direction des Systèmes d'Information), 6 ateliers d'une demi-journée à distance, organisés par thématique, ont permis d'examiner précisément les actions techniques à entreprendre. Pour l'équipe éditoriale, l'objectif à atteindre consistait à produire un contenu accessible (contributions et mises à jour du site, élaboration de documents accessibles proposés en téléchargement). Un guide des bonnes pratiques en matière éditoriale a été rédigé ; son objectif est de servir de "bible" aux équipes actuelles ou futures.

V-Technologies suit actuellement les équipes internes du Bas-Rhin dans la mise en œuvre des corrections sur le site (troisième étape), mais apporte aussi son conseil en matière de nouvelles prestations (préconisations dans de futurs cahiers des charges, validation de maquettes graphiques...)...

Sur la base du travail actuel et des documents fournis, le Conseil départemental dispose de tous les éléments lui permettant :

- de connaître l'état actuel de l'accessibilité du site, et l'écart par rapport à l'objectif de départ ;
- de comprendre la nature des corrections et d'en évaluer les impacts ;
- de disposer des éléments de décision concernant les priorités de correction, la répartition des tâches, et la feuille de route ;
- de mettre en œuvre une politique d'accessibilité numérique sur l'ensemble de ses projets, à plus long terme.

Nous contacter : <mailto:guillaume.lory@vtech.fr>



## Formation professionnelle : accessibilité pour les contributeurs

### Objectifs

Cette formation s'adresse aux personnels intervenant dans les phases de rédaction et de contribution de. Les **objectifs** de ces sessions seront notamment de :

- **Comprendre les enjeux** de la prise en compte de l'accessibilité numérique,
- **Savoir mettre en œuvre** l'accessibilité numérique dans les contenus rédactionnels **avec votre CMS**.

A l'issue de cette session, les participants auront acquis une compréhension précise de l'accessibilité, et de ses **impacts métier**. Enfin, ils auront aussi connaissance des ressources numériques disponibles sur le Web (référentiels, guides, vidéos, outils d'évaluation).

### Logistique

- Public : **rédacteur / contributeur**
- Nombre de participants max : **10**
- Equipement nécessaire : salle équipée d'un rétroprojecteur, d'ordinateurs (équipés des outils bureautiques et navigateurs web utilisés dans votre entité) et d'un accès au CMS pour l'ensemble des participants et le formateur.

### Formateurs

- Aurélien Levy
- Laurent Denis
- Eric Gateau

### Programme

- Intégrer l'accessibilité numérique dans les contenus rédactionnels (30min)

- **Les règles pour rédiger accessible (2h)**
  - Structurer son contenu (titre, liste, tableaux, citation)
  - Illustrer et enrichir son contenu (image, image légendée, multimédia)
  - Présenter son contenu (effet de mise en forme, alignement, couleur, typo)
  - Insérer des liens dans son contenu (poids / format / langue, intitulé)
  - Faciliter la compréhension de son contenu (langue, jargon, structure rédactionnelle, abréviations, longueur)
- **Produire un contenu accessible dans un CMS (2h)**
  - Le CMS allié ou ennemi ?
  - Les barres wysiwyg
  - La gestion des images
  - La gestion des vidéos
  - La gestion des pièces jointes
  - La gestion des langues
  - Le workflow de publication
  - Comment vérifier l'accessibilité de son contenu
  - Exercices de mise en œuvre / correction dans un ou plusieurs CMS
- **Produire des documents bureautiques accessibles (2h)**
  - Utilisation des styles et sommaire
  - Alternatives aux images
  - Métadonnées (titre, langue globale et changement de langue)
  - Tableaux de données,
  - zone de texte et multi-colonnage
  - Export pdf balisé
  - Accessibilité des pdf
  - Comment vérifier l'accessibilité de son document
  - Exercice de mise en œuvre / corrections
- **Bilan et échanges (30 minutes)**

### Plus d'information ?

Appelez nous au +33 (5) 56 401 402

Ou envoyez-nous un mail [contact@temesis.com](mailto:contact@temesis.com)

Temesis est spécialisée sur la qualité et l'accessibilité des sites Internet. Les experts Temesis ont produit les deux premières versions du référentiel général d'accessibilité pour les administrations, les référentiels Opquast, ont contribué au référentiel national de l'Etat du Luxembourg. Nous prôtons une vision pragmatique, inclusive et tolérante de l'accessibilité. Dans toutes nos missions et formations, nous veillons à tenir compte de vos contraintes et de votre contexte. Bien loin de l'approche accessibilité et conformité à tout prix, nous veillons à une efficacité maximum de nos interventions pour les personnes handicapées mais aussi pour tous les utilisateurs de votre sites Internet.

## NOS SOLUTIONS

### Savoir où en est mon site et me mettre en conformité avec la loi

- **Audit** par échantillonnage (RGAA, WCAG, AccessiWeb, etc.)
- **Suivi des corrections**
- Production de la **déclaration de conformité** RGAA

### Me faire accompagner pour choisir, concevoir et vérifier

- Revue de **cahier des charges et des spécifications**
- Assistance à la **qualification d'un outil ou d'une agence**
- **Recette** à chaque étape de la production (mini-audits successifs des maquettes ergonomiques, des maquettes graphiques, des intégrations HTML, etc.)

### Acquérir de nouvelles compétences

- Sensibilisation à l'accessibilité
- Savoir **vérifier l'accessibilité** d'un site avec le RGAA
- Savoir **intégrer l'accessibilité dans son projet**
- Savoir **produire un site accessible** et conforme RGAA

## NOS OUTILS

### Audit et pilotage de projet

Service en ligne permettant de **produire des rapports d'analyse**, de partager les résultats et de **piloter une démarche de mise en conformité**

<http://reporting.opquast.com>

### Analyse automatique

Opquast desktop est une **extension Firefox gratuite et open source** pour vérifier automatiquement des centaines de tests (RGAA, AccessiWeb, etc.)

<http://desktop.opquast.com>

## Références



### Mais aussi ...

*Présidence de la République, Ministères (Santé, Travail, Intérieur, etc.), Régions (Midi Pyrénées, Champagne Ardennes, Bretagne, Rhône, PACA, etc.), Conseils Généraux (18, 22, 31, 33, 44, 51, 56, 62, etc.), Défenseur des droits, CNSA, Agefiph, Radio France, BNF, Oseo, Sciences Po, Ademe, BRGM, Maaf, Groupama, Maif, AXA, LCL, BNP Personal finance, Société Générale, CIC, Schneider Electric, Arte, The Paciello Group, etc.*