

Domaine :	Recherche	Référencement
Niveau :	Pour tous	Avancé

Wordpress a évolué, au fil des ans et des différentes versions, d'un système de création de blogs à un outil CMS à part entière. Il permet donc de créer des sites web, de façon rapide, simple et efficace. A tel point qu'aujourd'hui, 10% des sites web l'utilisent. Aussi, l'optimisation d'un tel site pour les moteurs de recherche devient primordiale et la configuration de l'outil doit être prise en compte pour améliorer la visibilité dans les résultats des moteurs de recherche. Voici la seconde partie de notre article qui va vous expliquer comment tirer le meilleur parti de votre plateforme Wordpress en termes de SEO. Après une présentation générale de Wordpress et de son approche SEO le mois dernier, nous nous penchons ce mois-ci sur les lacunes du CMS et les solutions à y apporter pour les combler...

Comme expliqué dans notre précédent article, **WordPress** est un CMS en partie optimisé pour le référencement naturel. De base, il permet de se positionner assez facilement sur les requêtes que l'on cible. Mais même en l'utilisant à bon escient, il possède de grosses lacunes, que l'on peut heureusement combler.

La duplication de contenus et de liens

Duplication des ancres

En référencement naturel, une même URL présente plusieurs fois dans une page est nuisible, puisque Google ne prendra en compte que le 1er lien qu'il rencontre, en occultant complètement les suivants. Nous avons déjà observé ce comportement grâce à un test réalisé dans notre laboratoire SEO, mais c'est une information également retrouvée dans d'autres articles sur le référencement :

<http://blog.axe-net.fr/ancres-multiples-referencement-test-seo/>
<http://www.seomoz.org/ugc/3-ways-to-avoid-the-first-link-counts-rule>
<http://labo.seomix.fr/test-duplication-lien.php>

Même si on optimise les ancres de chaque lien, seul le 1er aura de l'importance. Et comme dans n'importe quel autre CMS, WordPress va lui aussi dupliquer les liens présents dans une même page, et cela à plusieurs niveaux :

- Dans le contenu même de la page ;
- Dans le thème ;
- Avec les plugins.

Pour ces trois niveaux de duplication, il existe une solution simple : ajouter une ancre # à la fin de chaque URL, ce qui va rediriger le visiteur vers une section prédéfinie de la page et non pas à son début. Par exemple, l'URL suivante redirige le lecteur non pas en haut de la page, mais bel et bien dans le footer, puisque l'ancre **#footer** existe dans le code :

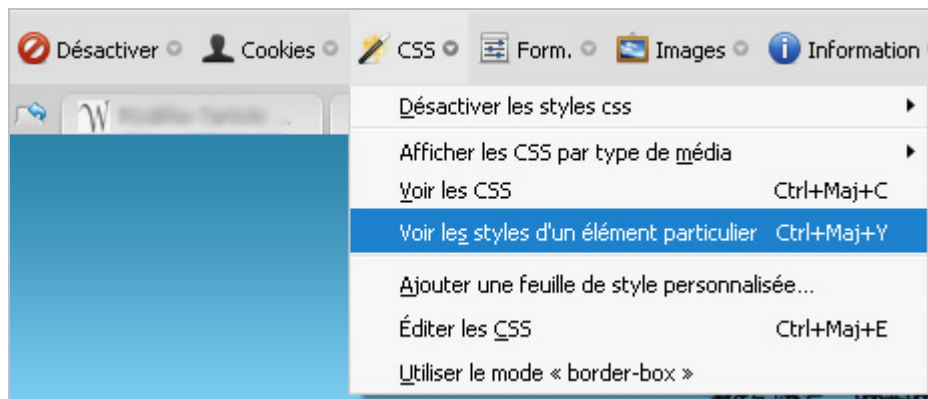
<http://www.seomix.fr/#footer>

Trouver une ancre

Pour connaître les ancres qui existent sur votre blog WordPress, nous conseillons d'utiliser l'extension **Webdeveloper** de Firefox, disponible ici :

<https://addons.mozilla.org/fr/firefox/addon/web-developer/>

Une fois installée, cliquez sur "CSS" => "Voir les styles d'un élément particulier".



Il suffit ensuite de survoler avec la souris les différentes parties de son blog pour voir apparaître les ancrs qui existent.



Dans cet exemple, on peut donc voir les ancre [#content](#), [#interieur](#) et [#articles](#).

A chaque fois que vous dupliquerez un lien, il faudra changer l'ancre utilisée. L'idéal est de cibler un sous contenu réel de votre article. Cela permet à Google de diviser votre article en sections homogène et ayant un sens propre, ce qui lui permet de faire remonter vos articles sur certaines requête. Par exemple, la requête "[Htaccess Compression](#)" fait remonter un article dédié au fichier HtAccess sur le site SeoMix, tout en donnant un accès direct à une section spécifique de son contenu (grâce aux ancrs bien définies).



Ancres dupliquées dans le contenu

En ce qui concerne le contenu même de vos articles, il est très simple de corriger le problème. Dès que vous ajoutez un lien dans votre contenu, veillez à y ajouter une ancre pour que Google la prenne en compte.



Ancre dupliquées dans le thème

Au niveau du thème, cela dépend fortement de celui qui est utilisé à la base. Là aussi, on va obligatoirement dupliquer des liens. En ce qui concerne le menu de navigation et le chemin de navigation, on ne peut pas y faire grand chose sans nuire à l'ergonomie. Mais en ce qui concerne les autres liens dupliqués, on peut agir.

Pour les ancres présentes dans le bloc d'information de l'article ("*Publié dans MaCatégorie*"), cherchez tout d'abord le code qui affiche la catégorie dans les fichiers suivants de votre thème WordPress :

[index.php](#) et/ou [home.php](#)
[single.php](#)
[tag.php](#)
[category.php](#)
[archive.php](#)
[author.php](#)
[nomdevotretaxonomie.php](#)
[nondevotrecustomposttype.php](#)
[loop-xxx.php](#)

En général, le code ressemblera à l'un de ces deux codes :

```
<?php the_category(',') ?>
```

```
<?php printf( __( '<span>Posted in</span> %2$s', 'twentyten' ), 'entry-utility-prep entry-utility-prep-cat-links', get_the_category_list( ', ' ) ); ?>
```

Remplacez-le par le code suivant :

```
<?php
    foreach((get_the_category()) as $childcat) {
        $parentcat = $childcat->category_parent;
        if( $parentcat != 0 ) echo " .get_cat_name($parentcat) . ' &raquo; ' ;}
    $category = get_the_category();
    $mycat = $category[0]->cat_name;
    $mycat2 = $category[0]->cat_slug;
    echo '<a href="'. $mycat2 .'/#votreancre' title="'. $mycat .'">'. $mycat . '</a>';
?>
```

Non seulement cela ajoute une ancre à votre lien ([/#votreancre](#)), mais cela rajoute les catégories parentes pour indiquer dans quelle sous-partie du site le visiteur va se diriger (ce qui rajoute du contenu texte et améliore l'ergonomie).

Pour les liens dupliqués dans un éventuel bloc "*Articles récents*", faites appel à ce code :

```
<ul>
<?php
    $result = $wpdb->get_results("SELECT comment_count,ID,post_title FROM $wpdb->posts WHERE comment_count >= 2 ORDER BY comment_count DESC LIMIT 0 , 4");
    foreach ($result as $topten) {
        $postid = $topten->ID;
        $title = $topten->post_title;
        $commentcount = $topten->comment_count;
        if ($commentcount != 0) { ?>
            <li><a href="<?php echo get_permalink($postid); ?>/#votreancre"
title="<?php echo $title ?>"><?php echo $title ?></a></li>
        <?php } } ?>
</ul>
```

Si cela était géré auparavant grâce à un widget, il faudra le désactiver puis ajouter ce hack dans le bon fichier, généralement dans :

[sidebar.php](#)
[rsidebar.php](#)

[lsidebar.php](#)

Libre à vous ensuite de le personnaliser pour l'adapter à votre charte graphique.

Pour les commentaires récents, ils ne sont généralement pas dupliqués, tout du moins pas dans le thème par défaut. Seulement, le fait d'y ajouter le lien renseigné par l'auteur est une magnifique incitation au spam, sans compter que cela vous fait perdre des visiteurs qui cliqueraient dessus au lieu de naviguer sur votre site. Nous vous conseillons donc de faire appel à ce code, qui indiquera toujours l'auteur du commentaire sans mettre son lien sur toutes les pages de votre site :

```
<?php
    $comment_array = $wpdb->get_results("SELECT comment_date_gmt,
comment_type, comment_author, comment_ID, comment_post_ID FROM $wpdb->comments
WHERE comment_approved = '1' AND comment_type != 'trackback' ORDER BY
comment_date_gmt DESC LIMIT 4");
    $comment_total = count($comment_array);
    echo '<ul>';
    for ($x = 0; $x < $comment_total; $x++) {
        echo '<li>'. $comment_array[$x]->comment_author.' dans ';
        echo '<a href="'. get_permalink($comment_array[$x]->comment_post_ID) .
'#comment-' . $comment_array[$x]->comment_ID . "'>';
        echo get_the_title($comment_array[$x]->comment_post_ID);
        echo '</a></li>';}
    echo '</ul>';?>
```

Ancre dupliquée dans les plugins

Pour terminer, certains plugins WordPress vont eux aussi créer des liens en doublons. Nous vous conseillons donc de les regarder un par un pour savoir quels sont ceux qui ajoutent des contenus dans vos articles ou dans votre thème, puis de faire en sorte de corriger ceux qui dupliquent vos liens.

On pense notamment à tous les plugins WordPress pour les contenus relatifs, qui vont systématiquement dupliquer vos liens. Nous expliquerons plus loin dans cet article le plugin que nous utilisons et comment le corriger...

Duplication des contenus textes

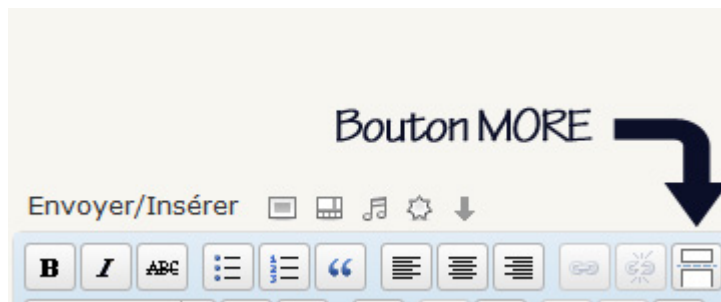
L'autre duplication de contenus qui existe concerne les doublons liés aux contenus des articles. Si on ne fait pas attention, on peut retrouver le même texte près de 6 fois sur un même blog :

- En page d'accueil ;
- Dans la page d'article ;
- Dans les archives par date ;
- Dans les pages d'auteurs ;
- Dans la catégorie ;
- Dans la page de tag.

La balise More

La 1ère méthode consiste à utiliser la balise `<!--more-->` lorsqu'on rédige un article. C'est ce qu'on appelle un délimiteur : tout ce qui se trouve après ne sera affiché que dans la page de l'article, tandis que l'article sera affiché sur les autres pages avec un bouton lire la suite. On élimine donc déjà une grosse partie du contenu dupliqué entre la page du contenu et le reste du site.

Pour ajouter la balise MORE dans vos contenus (et donc le bouton lire la suite), il suffit d'appuyer sur le bouton correspondant lorsque l'on rédige un article



Sur le thème par défaut Twenty Ten, voici ce à quoi ressemblera votre lien "Lire La suite" avec la balise More :

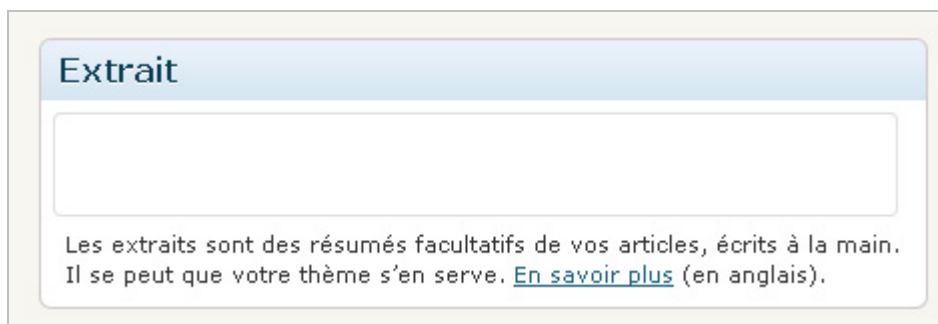


Excerpt et Content

Pour information, l'Excerpt est un résumé de votre article qu'il faut avoir rédigé manuellement (appelé *Extrait* en version française). Sur la nouvelle version de WordPress (en version 3.1), il est désactivé par défaut. Pour le faire apparaître, rendez-vous sur la page d'édition de votre article, puis cliquez en haut à droite sur "Options de l'écran" puis sur "Excerpt".



Vous aurez ensuite accès à ce bloc dans lequel vous devrez rédiger une description unique pour chaque article. L'idéal est d'écrire au moins 200 caractères (ce qui représente 3 à 4 phrases).



Pour réduire le duplicate content, on va donc jouer sur des variantes du texte affiché avant la balise MORE. Par défaut, WordPress affiche tout ce qui a été rédigé avant l'ajout de cette séparation, à condition que votre thème fasse appel à la fonction `<?php the_content() ?>`.

Pour enlever le duplicate content de l'accueil, utilisez :

`<?php the_content() ?>` en page d'accueil (fichiers `index.php` ou `home.php`)

`<?php the_excerpt() ?>` pour tous les autres fichiers.

Grâce à cette astuce, le seul contenu dupliqué restant est celui présent entre les différents types d'archives : catégories, tags, auteurs et dates, puisque la page d'article affiche tout le contenu et que l'accueil affiche le contenu avant le bouton *lire la suite*.

On va ensuite récupérer le contenu complet de l'article ou l'excerpt pour le tronquer, ce qui réduira le duplicate content. Nous vous conseillons de le faire pour les archives moins pertinentes, comme celles de l'auteur et des archives ([archives.php](#) et [author.php](#)). Dans le code ci-dessous :

- Vous pourrez varier la longueur en modifiant `185` par le nombre de caractères désiré.
- Vous pourrez varier pour chaque fichier le code utilisé : une fois l'excerpt, puis une fois le contenu.
- Remplacez `<?php the_content() ?>` ou `<?php the_excerpt() ?>` des fichiers cités par :
Avec l'excerpt :

```
<?php
    $len = 185;
    $newExcerpt = substr($post->post_excerpt, 0, $len);
    if(strlen($newExcerpt) < strlen($post->post_excerpt)) {$newExcerpt =
    $newExcerpt."...";}
    echo "<p>".$newExcerpt."</p>";
?>
```

Avec le contenu :

```
<?php
    $len = 185;
    $newcontent = substr($post->post_content, 0, $len);
    if(strlen($newcontent) < strlen($post->post_content)) {$newcontent =
    $newcontent."...";}
    echo $newcontent;
?>
```

Pour les archives par date, nous vous conseillons de bannir ces pages car cette hiérarchisation n'a tout simplement aucun sens en référencement naturel. Supprimez donc toute référence à celles-ci dans votre contenu, votre thème et dans vos widgets.

Des ajouts inutiles

Un header trop chargé

Par défaut, WordPress va charger plusieurs informations dans le *header* d'un blog. Cela pose problème à la fois pour la sécurité, pour le temps de chargement et pour le référencement naturel. La surcharge d'information va augmenter le poids de la page, sans en augmenter la pertinence. Cela reste minime, mais il n'existe pas de petites optimisations pour site. S'il est rapide, il sera meilleur pour le visiteur et le moteur de recherche. Parmi ce trop plein d'information, on retrouve :

- Des informations relatives au CMS et à sa version (ce qui représente le risque de sécurité dont nous parlions auparavant).
- Des liens vers différents contenus, sauf que ces liens dans un header ne servent à rien en référencement naturel (article suivant, article précédent, article le plus récent, ...)

```
<link rel='stylesheet' id='admin-bar-css' href='http://.../wp-includes/css/admin-bar.css?ver=20110122' type='text/css' media='all' />
<script type='text/javascript' src='http://.../wp-includes/js/l10n.js?ver=20101110'></script>
<script type='text/javascript' src='http://.../wp-includes/js/comment-reply.js?ver=20090102'></script>
<script type='text/javascript' src='http://.../wp-includes/js/jquery/jquery.js?ver=1.4.4'></script>
<script type='text/javascript' src='http://.../wp-content/plugins/fancybox-for-wordpress/js/jquery.fancybox-1.2.6.min.js?ver=1.3.2'></script>
<link rel='EditURI' type='application/rsd+xml' title='RSD' href='http://.../xmlrpc.php?rsd' />
<link rel='wlanifest' type='application/wlanifest+xml' href='http://.../wp-includes/wlanifest.xml' />
<link rel='index' title='Site de Demo' href='http://...' />
<link rel='start' title='A Simple Post with Text' href='http://.../blog/2008/08/02/a-simple-post-with-text/' />
<link rel='prev' title='An Ordered List Post' href='http://.../blog/2008/09/17/an-ordered-list-post/' />
<link rel='next' title='Bonjour tout le monde&nbsp;!' href='http://.../blog/2011/03/03/bonjour-tout-le-monde/' />
<meta name='generator' content='WordPress 3.1' />
<link rel='canonical' href='http://.../blog/2008/09/17/another-post-with-everything-in-it/' />
<link rel='shortlink' href='http://.../?p=56' />
<link rel='stylesheet' href='http://.../wp-content/plugins/debug-queries/css/style-frontend.css' type='text/css' media='screen' />
```

Pour désactiver le tout, rien de plus simple. Ajoutez le code suivant dans le fichier `functions.php` de votre thème WordPress :

```
remove_action('wp_head', 'wp_generator');
remove_action('wp_head', 'rsd_link');
remove_action('wp_head', 'wlanifest_link');
remove_action('wp_head', 'index_rel_link');
remove_action('wp_head', 'parent_post_rel_link');
remove_action('wp_head', 'start_post_rel_link');
remove_action('wp_head', 'adjacent_posts_rel_link_wp_head');
```

Dans votre fichier `header.php`, vérifiez si cette ligne est présente et supprimez-la :

```
<link rel="profile" href="http://gmpg.org/xfn/11" />
```

Trop de classifications différentes : tags, catégories et archives

Quel que soit votre paramétrage de WordPress, le point faible reste les pages de tags, de catégories et d'auteurs (celle par dates ont été désactivées comme indiqué précédemment).

Le problème qui se pose est multiple :

- La duplication de contenus (même si cela est en partie corrigé avec les astuces données précédemment).
- L'indexation de ces pages.

Pour les résoudre, voici quelques améliorations à apporter...

Les articles par page :

Commencez par augmenter le nombre d'article par page : cela réduira le nombre de pages à indexer, surtout qu'il s'agira à chaque fois de celles qui sont le moins pertinentes comme www.monsite.com/page/2/.

De plus, cela augmentera le contenu des pages de tags, et donc leur pertinence vis-à-vis du mot clé. On conseille donc d'avoir au minimum 8 à 10 articles par pages. A vous de trouver le juste milieu entre ergonomie utilisateur et augmentation de la pertinence pour les moteurs de recherche.

La pagination :

Deuxième point, la pagination. Par défaut, vous avez une navigation simpliste "Articles précédents" et "Articles suivants" en bas de page. Le problème est double : d'une part, il faudra beaucoup de clics pour le moteur de recherche pour aller indexer la dernière page du blog, d'autre part c'est anti-ergonomique pour l'utilisateur. Il faut donc changer cette navigation. Pour cela, nous vous conseillons le plugin PageNavi disponible ici : <http://wordpress.org/extend/plugins/wp-pagenavi/>.

Installez-le sur votre blog Wordpress, puis paramétrez-le comme suit :

Pour les champs texte :

Page %CURRENT_PAGE% sur %TOTAL_PAGES%

%PAGE_NUMBER%
%PAGE_NUMBER%
"1ère
Dernière"

Laissez les 4 derniers champs vides car ils nuisent à la lisibilité et ils créent des doublons de liens pour la page suivante et précédente.

Par défaut, utilisez ces valeurs pour les autres paramètres :

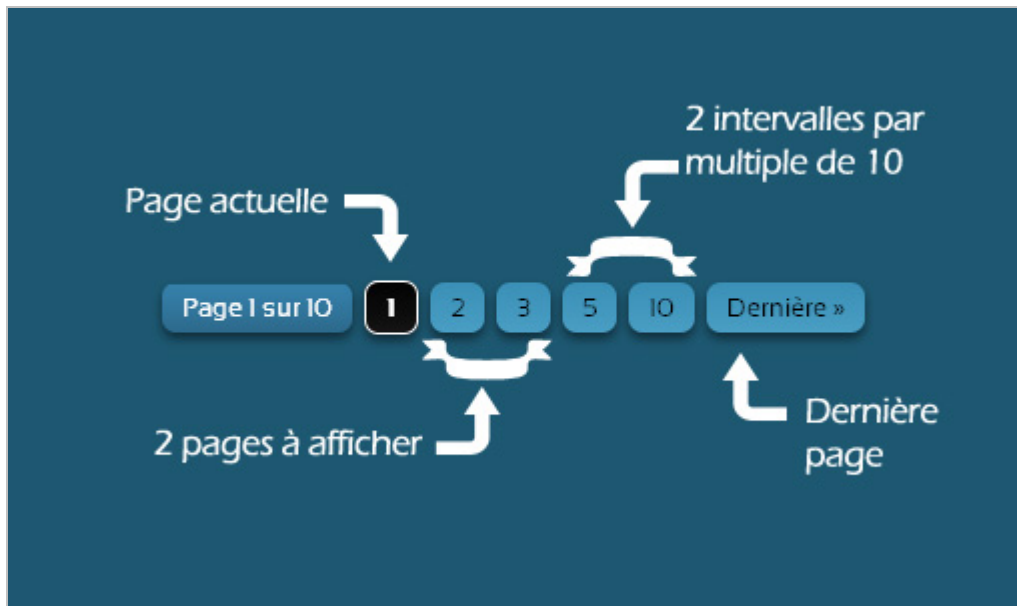
- Nombre de pages à afficher : 3 ;
- Nombre d'étapes intermédiaires : 3 ;
- Afficher les étapes intermédiaires par multiples de 5.

Texte pour le nombre de pages	Page %CURRENT_PAGE% sur %TOTAL_PAGES% <small>%CURRENT_PAGE% - Le numéro de la page active. %TOTAL_PAGES% - Le nombre total de pages.</small>
Texte pour la page affichée	%PAGE_NUMBER% <small>%PAGE_NUMBER% - Le numéro de page.</small>
Texte pour les pages	%PAGE_NUMBER% <small>%PAGE_NUMBER% - Le numéro de page.</small>
Texte pour la première page	« 1ère <small>%TOTAL_PAGES% - Le nombre total de pages.</small>
Texte pour la dernière page	Dernière » <small>%TOTAL_PAGES% - Le nombre total de pages.</small>
Texte pour la page précédente	
Texte pour la page suivante	
Texte pour les pages intermédiaires précédentes	
Texte pour les pages intermédiaires suivantes	
Options WP-PageNavi	
Use pagenavi-css.css	<input type="checkbox"/>
Style de présentation de la navigation	Normal
Toujours afficher la navigation WP-PageNavi	<input type="checkbox"/> Show navigation even if there's only one page.
Nombre de pages à afficher	3
Nombre d'étapes intermédiaires de pagination à afficher	3 <small>Les étapes intermédiaires de pagination sont utiles sur les sites comprenant un très grand nombre d'articles. For example, WP-PageNavi will display: Pages 1, 2, 3, 4, 5, 10, 20, 30, 40, 50. Mettez 0 pour ne pas inclure d'étapes intermédiaires de pagination.</small>
Afficher les étapes intermédiaires de pagination en multiples de	5 <small>Par exemple, si vous choisissez des étapes multiples de 5, WP-PageNavi affichera : 5, 10, 15, 20, 25...</small>

Attention au niveau des trois derniers points : ils vont dépendre de la taille de votre blog.

Sur un petit blog (moins de 100 articles) ayant 8 à 10 articles par pages, ces réglages sont parfaits. Mais si votre blog en contient des centaines voir des milliers, il va falloir paramétrer le tout autrement. Tout d'abord, il faudra augmenter le nombre d'articles par page, avec par exemple 15 ou 20 articles voir plus.

Voici le rendu de notre paramétrage pour un blog de (seulement) 10 pages :



Si votre site a beaucoup de pages, il faut faire un peu de calcul mental. L'idée est que Google doit parvenir à n'importe quelle page de votre site en moins de 5 clics pour faciliter au maximum l'indexation. Notre méthode de calcul est un peu complexe mais a fait ses preuves :

- On divise par deux le nombre de pages du blog ;
- On divise ensuite ce chiffre par 3 ou 4 (c'est-à-dire le nombre d'intervalles), ce qui nous donne le multiple à utiliser pour afficher les étapes intermédiaires ;
- Divisez ensuite l'intervalle par deux, et divisez de nouveau le tout par 2 pour obtenir le nombre de pages suivantes à afficher.

Cela nous donne donc l'enchaînement suivant:

- Nombre d'intervalle = 3 à 5 (3 pour un petit blog, 4 voir 5 pour les plus gros) ;
- Multiple d'intervalle = Nombre de pages total / 2 / Nombre d'intervalle ;
- Nombre de pages à afficher = Multiple d'intervalle / 2 / 2.

Prenons l'exemple d'un blog ayant 178 pages, et que l'on veuille indexer la 89ème page.

1er clic : Page d'accueil du site

2ème clic : arrivée sur la page 80 (Multiple d'intervalle de 20 avec 4 nombres d'intervalles)

3ème clic : arrivée sur la page 86 (Nombre de page à afficher à 6)

4ème clic : arrivée sur la page 89

En moins de 4 clics, on arrive donc à atteindre n'importe quelle page du site. Attention cependant, il peut arriver que sur certains blogs monstrueux on obtienne un nombre de page à afficher trop important, auquel cas on n'aura pas le choix que d'avoir un paramétrage qui permettra d'atteindre chaque article en 5 clics au lieu de 4.

Les archives dates :

Pensez à supprimer définitivement les archives par date, du type monsite.fr/blog/2011/03/. Nous l'avons expliqué au début de cet article : enlevez de votre thème tous les références aux archives par dates, souvent *via* un widget ou dans le fichier [sidebar.php](#).

C'est cependant insuffisant. Il faut forcer la désindexation de ces contenus. Vérifiez donc que le plugin que vous utilisez pour le sitemap n'inclue pas les pages de dates. Ajoutez également ces lignes à votre robot.txt, en répétant l'opération pour chaque année :

Disallow: /2010/

Disallow: /2011/

...

Les tags et les articles :

Enfin, pensez toujours à limiter le nombre de mots clés par article. On ne le répétera jamais assez. Il faut réussir à sélectionner les mots clés de chaque article. Le mieux est d'en avoir 2 à 4. Ils doivent être choisis selon leur pertinence vis-à-vis de l'article et du potentiel de trafic qu'ils peuvent générer. Pour cela, vous pouvez notamment utiliser ces outils :

- **Google Insight** : <http://www.google.com/insights/search?hl=fr#>

- **Adwords External Tool** : <https://adwords.google.com/select/KeywordToolExternal>

- Google et l'option de recherche **roue magique** ou **recherche associée** (dans la colonne de gauche quand on a effectué une recherche).

Par exemple, si on devait choisir des mots clés pour cet article d'Abondance, nous choisirions : "Référencement WordPress" - "Optimisation WordPress" - "Duplicate Content".

Des fonctionnalités manquantes

Maintenant que nous avons fait le tour des problèmes de duplications et d'indexation d'un blog WordPress, passons aux autres lacunes de ce CMS pour le SEO.

Le fichier sitemap

Parmi les oublis majeurs, il y a le fichier sitemap, qui facilite l'indexation des contenus d'un site par Google, surtout pour ceux ayant beaucoup de contenus. Cela n'améliorera pas le positionnement, mais va contribuer à la bonne compréhension de la structure d'un site et à sa complète indexation. Nous conseillons l'utilisation d'un plugin comme **Google XML Sitemap** : <http://wordpress.org/extend/plugins/google-sitemap-generator/>

Au niveau du paramétrage, c'est relativement simple :

- Demandez la création d'un sitemap normal et d'un autre Gzipé ;
- Cliquez sur *informer* pour chaque moteur de recherche (il faudra demander une clé API pour Yahoo) ;
- Dans *Contenu du Sitemap*, il faut tout cocher sauf *Archives* (dont on veut se débarrasser), et *author* si vous ne souhaitez pas créer de pages dédiées aux auteurs.

Toutes les autres options du plugin ne servent à rien. L'important, c'est de lister toutes les pages du site dans le fichier avec les informations relatives (catégories, date de publication, ...), ce que fera très bien ce plugin.

Pour ceux qui veulent apparaître dans Google Actualités, on peut faire appel au plugin **Google News Sitemap** : <http://wordpress.org/extend/plugins/google-news-sitemap/>

Pensez pour terminer à vous rendre dans les différents centres webmaster de Google, Yahoo et Bing, pour y ajouter vos sitemaps.

Les articles relatifs

Là aussi, c'est un gros oubli des développeurs WordPress. La gestion des articles relatifs est un point essentiel pour le maillage interne d'un blog, surtout pour les gros sites où il sera trop long de revenir sur chaque contenu pour refaire des liens internes après chaque publication.

La bonne gestion de ce maillage entre contenus similaires renforce le poids du contenu, et favorise donc le positionnement des pages.

Pour WordPress, il n'existe pas de hack suffisamment efficace, qui prenne en compte à la fois le titre, les tags, la catégorie et le contenu d'un article. On fera donc appel au plugin **YARRP (Yet Another Related Post Plugin)** disponible ici : <http://wordpress.org/extend/plugins/yes-another-related-posts-plugin/>.

Le plugin se configure en 4 blocs. Le 1er ne sert à rien ("*Le Corpus*"). Dans le second ("*Options de complémentarité*"), nous conseillons de configurer un seuil de 3 ou 4 minimum

pour que les articles relatifs soient un minimum pertinent. Plus on aura d'articles sur son blog, plus on pourra augmenter cette valeur (il faudra tester). Nous vous conseillons de cocher ensuite la prise en compte de chaque élément, et de prendre en compte avec plus d'importance le titre des articles, ainsi que d'activer la complémentarité croisée entre articles et pages.

Dans le troisième bloc, choisissez d'utiliser un modèle personnalisé plutôt que d'afficher automatiquement le plugin. Cela permettra de la placer réellement où vous le désirez, mais surtout de modifier les ancres pour éviter la duplication.

On va ensuite devoir faire quelques manipulations dans le thème :

- Cherchez ce répertoire : www.votresite.com/wp-content/plugins/yet-another-related-posts-plugin/yarpp-templates.

- Copiez tous les fichiers que vous y trouverez dans le répertoire de votre thème (donc dans www.votresite.com/wp-content/themes/nomdutheme). Vous devrez donc avoir copié ces fichiers :

[yarpp-template-example.php](#)
[yarpp-template-list.php](#)
[yarpp-template-photoblog.php](#)
[yarpp-template-random.php](#)
[yarpp-template-thumbnail.php](#)
[yarpp-template-wpml.php](#)

- Ensuite, copiez/collez ce code dans le fichier [single.php](#) de votre thème pour pouvoir afficher le bloc des articles relatifs :

```
<?php if (function_exists('related_posts')) { echo related_posts(); } ?>
```

- Revenez dans les options du plugin, et choisissez l'un de ces templates dans la liste déroulante (toujours dans le 3ème bloc). Nous vous conseillons de les essayer tous afin de savoir lequel s'adapte le mieux à votre charte graphique. Une fois que vous aurez trouvé celui qui vous convient, modifiez-le pour rajouter une ancre de votre choix. Le code initial des liens doit ressembler à cela :

```
a href="<?php the\_permalink\(\) ?>"
```

Modifiez-le-en:

```
a href="<?php the\_permalink\(\) ?>#votreancre"
```

Au niveau du flux RSS (dernier bloc), il faut bien avouer que nous n'en sommes pas fan : si vous l'activez, au mieux les agrégateurs de contenus vont vous rajouter des liens supplémentaires, mais cela va fortement nuire à l'expérience utilisateur quand ils voudront lire vos actualités. Nous conseillons donc de désactiver cette option.

Une table des matières

Un autre manque de WordPress est la gestion d'un sommaire ou d'une table des matières. Si vous rédigez des articles courts, l'intérêt est minime. Par contre, si vous rédigez des contenus longs, cela s'avère être une arme redoutable !

Basé sur les titres H2 à H4 et des ancrs, un sommaire peut fortement aider Google à diviser en sous partie une page. Cela permet ainsi de ressortir plus facilement dans le moteur de recherche lorsque la requête de l'internaute ne concerne qu'une sous partie d'un contenu. Cela va donc de pair avec l'exemple pris en début d'article sur une requête qui dirige l'internaute vers une sous partie d'un de mes articles.

Pour réaliser ce sommaire automatiquement, nous passons tout simplement par un hack à coller dans le fichier [functions.php](#) de votre thème :

```
// ***** Fonction de table des matières - Modifié par Daniel Roch - Original par Scott Yang
add_shortcode("toc", "removetoc");
function removetoc(){return"";}
class YLSY_TableOfContents {
    function add_toc($level, $tocid, $text) {
        $this->toc[] = array($this->pagenum, $level, $tocid, $text);
        $this->minlevel = min($this->minlevel, $level);}
    function get_tocid($text) {
        $text = sanitize_title_with_dashes($text);
        $text = preg_replace('|%([a-fA-F0-9][a-fA-F0-9])|', "", $text);
        $tocid = $text;
        $count = 0;
        while (isset($this->tocmap[$tocid]))
            $tocid = $text.strval(++ $count);
        $this->tocmap[$tocid] = true;
        return "$tocid";}
    function get_toc() {
        global $wp_rewrite;
        if (isset($this->toccache))return $this->toccache;
        if (!isset($this->toc))return "";
        $html .= '<ul>';
        $permalink = get_permalink($this->postid);
        for ($i = 0; $i < sizeof($this->toc); $i++) {
            list($pagenum, $level, $tocid, $text) = $this->toc[$i];
            $link = $permalink;
            if ($pagenum > 1) {
                if ($wp_rewrite->using_permalinks()) $link =
                trailingslashit($link)."$pagenum/";
                else $link .= "&page=$pagenum";}
            $link = "<a href=\"$link#$tocid\">$text</a>";
            if ($i == 0) {
                $level = min($level, $this->minlevel);
                $stack = array($level);
                if ( $level == 2) {$one='&raquo;';}if ( $level > 2)
                {$one='&rArr;';}
                $html .= "<li class='toc$level'>$one $link";
            } else {
                $prev = $stack[sizeof($stack)-1];
                if ($level == $prev) {
                    if ( $level == 2) {$one='&raquo;';}if ( $level > 2)
                    {$one='&rArr;';}
                    $html .= "</li>\n<li class='toc$level'>$one $link";
                } elseif ($level < $prev) {
                    while (sizeof($stack) > 1) {
                        array_pop($stack);
                        $html .= "</li>";
                        $prev = $stack[sizeof($stack)-1];
                        if ($level >= $prev)break;}
                    if ( $level == 2) {$one='&raquo;';}if ( $level > 2)
                    {$one='&rArr;';}
                }
            }
        }
    }
}
```

```

        $html .= "</li>\n<li class='toc$level'>$one $link";
    } else {
        $stack[] = $level;
        if ( $level == 2) {$one='&raquo;';}if ( $level > 2)
    {$one='&rArr;';}
        $html .= "\n<li class='toc$level'>$one $link";}}
    while (sizeof($stack) > 0) {
        array_pop($stack);
        $html .= "</li>";}
    $final = "";
    $html .= '</ul>';
    if ($html != "" )
        $final = "<div><div>Dans cet article</div><div
class=\"toc\">\n".$html."</div></div>";
    return $final;}
function replace_heading($match) {
    if ($match[0] == '<!--nextpage-->') {
        error_log('next');
        $this->pagenum ++;
        return $match[0];}
    $tocid = $this->get_tocid($match[3]);
    $this->add_toc(intval($match[1]), $tocid, $match[3]);
    return "<h$match[1] id=\"\$tocid\"$match[2]>$match[3]</h$match[1]>";}
function the_content($content) {
    $this->toc = array();
    $this->tocmap = array();
    $this->tocache = null;
    $this->minlevel = 6;
    $this->pagenum = 1;
    $regex = '#<h([1-6])(.*?)(.*?)</h\1>|<!--nextpage-->#';
    $content = preg_replace_callback($regex,
    array(&$this, 'replace_heading'), $content);
    return preg_replace('|(<p>)?<!--TOC-->(</p>)?|', $this->get_toc(),
    $content);}
function the_posts($posts) {
    for ($i = 0; $i < sizeof($posts); $i ++ ) {
        $post = &$posts[$i];
        $this->postid = $post->ID;
        $post->post_content = $this->the_content($post->post_content);
        $post->post_toc = $this->get_toc();}
    return $posts;}
};
add_filter('the_posts', array(new YLSY_TableOfContents, 'the_posts'));

```

Copiez ensuite le code ci-dessous dans le fichier [single.php](#) de votre thème pour le faire apparaître. Si votre article contient des titres h2, h3 ou h4, le sommaire sera automatiquement généré.

```
<?php echo $post->post_toc; ?>
```

Le code est de Scott Yang, et a été modifié par nos soins pour un peu plus de clarté. Sans mise en page, voici le rendu par défaut :

Dans cet article

- » [Level 2 Heading](#)
- ⇒ [Level 3 Heading](#)
- ⇒ [Level 4 Heading](#)
- ⇒ [Level 5 Heading](#)
- ⇒ [Level 6 Heading](#)
- ⇒ [An Unordered List](#)
- ⇒ [Image with no alignment](#)
- ⇒ [An Ordered List](#)

Une page plan de site

En complément d'une bonne structure, une page plan de site permet à Google de facilement indexer l'intégralité du contenu d'un site Internet. Pour en créer une, il faudra passer par une modification de votre thème. Nous vous suggérons donc deux tutoriels :

- Une page plan de site simple mais efficace : <http://wpchannel.com/creer-sitemap-plugin-wordpress/> ;
- Une page plan de site graphique et plus complexe (gestion des catégories et sous catégories) : <http://www.seomix.fr/wordpress/themes/plan-de-site-sitemap-dynamique/> ;

Les redirections 301

WordPress gère mal les redirections 301. Quand on renomme l'URL d'un article, il parvient à faire la corrélation et à rediriger. Dans les autres cas, c'est l'impasse. Pour pallier le manque, nous vous conseillons le plugin **Redirection** disponible ici : <http://wordpress.org/extend/plugins/redirection/>

Son utilisation est simple. Dans *Outils* => *Redirection*, vous pourrez mettre en place toutes les redirections 301 désirées.

Ajouter une nouvelle redirection

URL source:

Correspond à: Action: Expression régulière:

URL cible:

Le chemin de navigation

Aussi appelé breadcrumb ou fil d'Ariane, le chemin de navigation sert à la fois pour l'utilisateur et pour les le référencement naturel : cela rajoute des liens directs vers la catégorie et les catégories parentes.

Pour en ajouter un facilement, utilisez tout simplement le plugin **BreadCrumb NavXt** : <http://wordpress.org/extend/plugins/breadcrumb-navxt/>

Pour conclure

WordPress est sans doute, à notre avis, le meilleur CMS actuel en ce qui concerne la prise en compte du référencement naturel. Si vous avez mis en place ces conseils et ceux de notre précédent article sur Abondance (février 2011), vous devriez avoir une véritable machine de guerre pour le SEO, à condition de rédiger des contenus pertinents et de créer de nombreux liens entrants de qualité sur votre site.

Daniel Roch, *Consultant WordPress, Référencement et Webmarketing chez SeoMix*
(<http://www.seomix.fr/>)

Réagissez à cet article sur le blog des abonnés d'Abondance :

<http://blog-abonnes.abondance.com/2011/03/wordpress-un-cms-optimise-pour-le.html>