

Bonnes pratiques de SEO ('white hat techniques')

par Guillaume Rossolini ([Tutoriels Web / SEO / PHP](#)) ([Blog](#))

Date de publication : 11 octobre 2006

Dernière mise à jour :




Le référencement est une sorte de science complexe. Les moteurs de recherche ne donnent pas toutes les précisions sur le fonctionnement de leurs algorithmes, ce qui laisse aux spécialistes du référencement la charge de deviner la plupart des paramètres. Les techniques approuvées par les moteurs de recherche portent généralement le nom de "white hat" (soit "chapeau blanc", une référence à l'honnêteté du webmestre en question).



I - Introduction.....	3
I-A - Remerciements.....	3
I-B - Préambule.....	3
I-C - Historique succinct.....	3
I-D - Objectif de cet article (en tant que webmestre).....	3
I-E - Le champ sémantique.....	4
II - Rédigez correctement votre code HTML.....	5
II-A - Première approche.....	5
Introduction.....	5
Exemple.....	5
II-B - Les liens : <a>.....	6
Première approche.....	6
Les liens dynamiques.....	7
II-C - Les images :	8
II-D - Les tableaux : <table>.....	9
II-E - Les divisions : <div>.....	9
II-F - Les listes :	10
II-G - Les formulaires : <form>.....	10
II-H - De manière générale.....	10
III - Utilisez les technologies à bon escient.....	12
III-A - Les images.....	12
III-B - Flash.....	12
III-C - AJAX.....	12
III-D - Les feuilles de style.....	12
III-E - Les headers, les redirections.....	13
III-F - Contrôlez les liens sortants.....	13
IV - Quelques outils.....	14
IV-A - Pourquoi utiliser des outils ?.....	14
IV-B - Les statistiques.....	14
IV-C - La publicité.....	14
V - Conclusion.....	15
V-A - Épilogue.....	15
V-B - Liens.....	15
V-C - L'auteur.....	15


I - Introduction

I-A - Remerciements

I-B - Préambule

Comme dans la plupart de mes articles, il pourra m'arriver de parler de " Google" alors que je parle bel et bien de l'ensemble des  **moteurs de recherche**. La raison est simple : c'est Google qui a introduit toute cette vision des choses et qui est la cause du remaniement total des techniques de  **SEO**.

Je vais commencer par tenter de cerner l'objectif des moteurs de recherche. Cela nous aidera à oeuvrer dans la bonne direction. Nous aborderons ensuite quelques techniques connues sous l'appellation "chapeau blanc" (*white hat*), c'est-à-dire qu'il s'agit de méthodes conseillées par les moteurs de recherche eux-mêmes. Il s'agit tout simplement d'optimiser un site Web pour répondre à une demande précise : le  **crawler** de  **référencement**. Ces techniques s'opposent au "chapeau noir" (*black hat*) qui feront l'objet d'un autre cours.


 *Tout ce que je dirai ici sera orienté vers la SEO pour votre site. Il n'est pas impossible que certains conseils entrent en conflit avec ce que vous avez pu lire ailleurs concernant des "bonnes pratiques de programmation". À vous de faire les bons choix.*

I-C - Historique succinct

J'ai déjà évoqué le contexte du Web à l'époque de l'apparition de Google. C'était un peu la foire, chaque webmestre faisait comme bon lui semblait et certains n'avaient aucun scrupule. Cela a conduit à une évolution des moteurs de recherche. Ils étaient alors très semblables à de simples répertoires puisqu'ils classaient les sites à partir d'informations fournies par le webmestre du site. L'avantage de cette méthode est la facilité de mise en place. L'inconvénient est que le webmestre peu scrupuleux a un contrôle presque total sur le positionnement de son site dans les résultats d'une recherche.

Il lui suffisait presque de dupliquer le contenu de son site dans le `<meta name="keywords" />` pour que le moteur de recherche, qui confiait en cette balise, fasse apparaître le site pour chacun de ces mots clefs. À l'aide de plusieurs techniques similaires, les moteurs de recherche d'antan étaient aisément dupés.

Google a introduit une notion très intéressante lors de son arrivée : son objectif a été de classer les sites non pas à l'aide uniquement de paramètres que le webmestre peut contrôler, mais également à l'aide d'éléments sur lesquels il ne peut agir que de manière insignifiante.


Il s'agit des  **backlinks**. J'ai suffisamment disserté là-dessus (*cf. mes autres articles*), je ne réitérerai pas mes propos.

I-D - Objectif de cet article (en tant que webmestre)

Notre objectif sera de nous concentrer sur les éléments de notre site en lesquels Google peut encore avoir **confiance** et qui peuvent donc nous servir à le guider convenablement.

Je crois fermement que le plus important ici est de construire une relation de confiance entre Google et notre site. Avec le temps, si notre site n'enfreint pas les règles de l'art et que le moteur de recherche estime que notre site est digne de confiance, il n'est pas exclu que Google offre des bonus au niveau du référencement.

En tant que webmestres, notre intérêt est donc de guider les bots des moteurs de recherche vers ce qu'ils cherchent en leur montrant un chemin correctement fléché.

 *Un élément peut ne pas suffisamment transparaître dans cet article et je souhaite en parler ici : **votre site Web est destiné à des êtres humains, pas à un robot**. À ce titre, le robot de référencement cherche toujours à trouver du sens dans vos pages. Il cherche la sémantique où il en est capable et traque les incohérences dans la mesure du possible.*


Soyez la référence : travaillez le contenu de votre site avant toute chose.

I-E - Le champ sémantique

La sémantique est un terme de la plus haute importance en informatique. Lorsqu'il s'agit d'écriture de code source, une bonne sémantique signifie à la fois que vous avez structuré votre code d'une manière intelligible et que vos noms de variables/fonctions/etc. sont totalement transparents. Un tel code source a très peu besoin de documentation.

Ici, il s'agit de déterminer le champ sémantique de votre site Web.

Cette étape est fondamentale, dans la mesure où tout s'orientera en fonction de ce choix. Sachez que les moteurs de recherche ont tendance à préférer trouver un site qui traite d'un thème unique. Pourquoi ? Parce que les internautes fonctionnent sur le même principe. De plus, votre nom de domaine devrait être un résumé de votre site.

 *Pour obtenir un bon référencement, je vous suggère de sélectionner soigneusement deux ou trois mots et de les utiliser dans la majorité de vos textes d'ancrage (les liens). Demandez à vos partenaires de faire de même lorsqu'ils vous offrent un lien sur leurs pages.*

Concentrez-vous sur quelques mots clefs phares, votre référencement sur ces mêmes mots s'en portera d'autant mieux. Bien sûr, si ces mots clefs correspondent au champ sémantique principal de votre site, les robots indexeurs vous rendront vos efforts. Évidemment (bis), choisissez des mots clefs à votre mesure : si vous penchez votre dévolu sur "apache" et "php" pour un site sur le développement Web, autant dire que vous aurez bien du mal à parvenir en première page des résultats sur ces deux mots clefs...

Essayez également de trouver des partenaires pour votre site afin de compléter votre thématique. Attention, il ne s'agit pas de concurrents mais bien de sites complémentaires.


II - Rédigez correctement votre code HTML

II-A - Première approche

Introduction


J'ai toujours détesté les livres emplies de fautes d'orthographe. Je n'étais pas encore au collège que j'y corrigeais les erreurs de français. Cela m'a toujours agacé, au point parfois de me faire abandonner ma lecture.

D'autre part, il existe les écrits expérimentaux : un poème écrit sans ponctuation (Guillaume Apollinaire, *Le Pont Mirabeau*, *Alcools*, 1913), un roman ne contenant pas une seule fois la lettre "e" (Georges Perec, *La disparition*, 1969), etc. Ces méthodes d'écriture ne sont pas destinées à être reproduites au quotidien. Elles sont rares. (1)
Corollaire : la grammaire française est régie par des règles auxquelles nous sommes habitués. Il est généralement déroutant de se trouver face à une situation qui ne les respecte pas. (2)

Suivant le même principe, le code  **HTML** d'une page Web doit être propre. Un code empli d'erreurs est signe que le webmestre s'occupe peu des détails. Je ne sais pas dans quelle mesure Google vérifie le code d'une page (je ne sais d'ailleurs pas s'ils le font) mais je sais qu'ils utilisent des bots. Pour avoir moi-même écrit des bots à plusieurs reprises, je sais à quel point il est frustrant de devoir pallier les erreurs ou les différences de conventions. Prenez par exemple un simple lien HTML `` : les guillemets peuvent être remplacés par des apostrophes ou bien être omis. Selon le navigateur que vous utilisez, il est probable qu'utiliser un guillemet et une apostrophe dans la même balise puisse également donner un résultat satisfaisant. Cette souplesse contribue à la facilité de créer des pages Web mais également à la complexité de certaines tâches : un crawler devra gérer toutes les situations...

De même que vous écrivez du code syntaxiquement correct, prenez garde à sa sémantique. Oui, le code HTML a une sémantique : essayez d'utiliser les balises HTML à bon escient. Par exemple, lorsque vous utilisez la balise `<table>`, le bot s'attendra à trouver des données à l'intérieur, par opposition à "des paragraphes de texte". Si vous souhaitez simplement diviser votre page, la balise adéquate est `<div>`.

La structure du document HTML reflète l'organisation de son contenu. C'est bête à dire mais c'est pourtant peu respecté.

 *Tout au long de ce cours, j'utiliserai des noms de classes, fichiers etc. répondant à une certaine syntaxe. C'est l'objet d'un autre tutoriel, je me bornerai ici à le résumer : en anglais, en minuscules et uniquement des lettres et le trait d'union.*

Exemple

En bref, il est recommandé de suivre les standards du Web. N'utilisez pas des balises contre leur nature. J'ai vu des gens chercher à faire défiler du texte coloré dans une balise `input type="text"` : c'est possible mais est-ce souhaitable ? La balise `input` est exclusivement réservée à une information provenant de l'utilisateur. N'essayez pas de l'utiliser pour produire des effets visuels... Votre site favori devrait être **w3.org**.

Voici deux exemples utilisant chacun une expression régulière afin d'extraire les liens d'une page Web. Le premier exemple permet de récupérer les liens correctement formés (ne disposant d'aucune option), tandis que le second tente de récupérer les liens (ne disposant d'aucune option), qu'ils soient bien ou mal formés.

Récupérer des liens simples

```
<?php
$regex = '#<a href="([^\"]*)">(.*?)</a>#Usi';
$html = file_get_contents('http://g-rossolini.developpez.com/');

if(preg_match_all($regex, $html, $matches, PREG_SET_ORDER)){
    echo '<pre>';
```

Récupérer des liens simples

```
print_r($matches);
echo '</pre>';
}
?>
```

Récupérer des liens mal écrits

```
<?php
$regex = '#<a href=["\']?([>]*)["\']?>(.*</a>#Usi';
$html = file_get_contents('http://g-rossolini.developpez.com/');

if(preg_match_all($regex, $html, $matches, PREG_SET_ORDER)){
    echo '<pre>';
    print_r($matches);
    echo '</pre>';
}
?>
```

L'expression régulière est visiblement plus complexe pour le second exemple.



Ce deuxième exemple est incomplet. Pour extraire correctement les informations de liens mal écrits, je ne vois pas de solution réellement exploitable. Il faudrait peut-être extraire le lien entier, puis le parcourir caractère par caractère en essayant de déterminer le début et la fin de chaque paramètre. J'imagine qu'il suffirait de regarder le code source d'un navigateur Web pour obtenir une solution fiable, mais ce n'est pas notre objectif ici.

Je pense que mon premier but est atteint : je viens de démontrer que cela devient rapidement un casse-tête si le code HTML n'est pas correct. C'est pour cette raison que les moteurs de recherche préfèrent les sites au code propre.

II-B - Les liens : <a>

Première approche

Dans la mesure où ils sont le fondement des algorithmes de référencement, les liens (ancres HTML) ont besoin d'une mention particulière.


Une balise de lien est composée de plusieurs attributs. Le plus courant est *href*, puisqu'il permet de donner une destination au lien. Dans mes exemples précédents, c'est le seul attribut présent.

Dans une situation réelle, l'attribut *title* joue un rôle important lors du référencement : il permet au navigateur d'afficher une infobulle d'information. C'est très utile pour associer des mots clefs à un lien dont le texte d'ancrage n'a rien à voir. Par exemple, le texte d'ancrage "cliquez ici" n'a aucun sens pour le moteur de recherche : il ne sait pas quels mots clefs associer au lien.

Un lien correctement formé est donc constitué principalement d'une URI (propriété *href*) ne contenant pas de caractères exotiques et, à défaut d'un texte d'ancrage significatif, d'un titre (propriété *title*) permettant de le mettre en relation avec des mots clefs.

Un lien doit toujours être associé à des mots clefs (soit par son texte d'ancrage soit par son titre) lorsqu'il a une URI de destination.

Il est possible d'introduire des mots clefs dans l'URI de destination afin de préciser davantage au moteur de recherche ce qu'il trouvera dans le document cible. C'est fondamental car cela permet éventuellement au moteur de recherche d'orienter son analyse sémantique autour du champ sémantique de ces mots clefs.

 Ici, il est bon de préciser qu'un moteur de recherche effectue une analyse sémantique des documents qu'il indexe.

Cette analyse permet plusieurs choses :

- Deviner le champ sémantique de la page afin de déterminer les mots clefs les plus appropriés ;
- Savoir si la page en question traite d'un thème fixe ou bien si les thèmes sont variés ;
- Repérer les pages ayant du contenu similaire et celles ayant du contenu dupliqué ;
- etc.

Pour continuer sur les liens, je vous propose quelques exemples permettant de résumer les options qui nous sont offertes.

Lien avec un ancrage qui n'est pas utilisé comme mot clef : la propriété 'title' est fortement recommandée

```
<a href="http://g-rossolini.developpez.com/tutoriels/php/expressions-regulieres/"
  title="Tutoriel sur les expressions régulières"
>Voici</a>
un tutoriel sur les expressions régulières.
```

Lien avec un ancrage utilisé comme mot clef : la propriété 'title' est facultative

```
Voici un <a href="http://g-rossolini.developpez.com/tutoriels/php/expressions-regulieres/"
  >tutoriel sur les expressions régulières</a>.
```

Autre lien avec un ancrage utilisé comme mot clef : la propriété 'title' est utile

```
Voici un <a href="http://g-rossolini.developpez.com/tutoriels/php/expressions-regulieres/"
  title="Cours sur les expressions rationnelles"
  >tutoriel sur les expressions régulières</a>.
```

Dans le dernier cas, je crains que la répétition de mots clefs (identiques) ne soit néfaste. En effet, certains d'entre eux sont situés à trois endroits : dans les propriétés *href* et *title* ainsi que dans le texte d'ancrage. Il est possible que cela soit vu comme du martèlement de mots clefs, c'est-à-dire une lourde insistance de la part du webmestre. Dans la mesure du possible, il faut essayer d'éviter cette situation car il n'est pas improbable que les moteurs de recherche pénalisent le classement de la page.

Cependant, la propriété *title* permet d'insérer quelques synonymes dans certaines situations, il ne faut donc pas la négliger.

Les liens dynamiques

Dans le cas des sites Web dynamiques, nous sommes fréquemment confrontés à des paramètres GET.

Exemple d'URI avec un paramètre

```
http://g-rossolini.developpez.com/tutoriels/php/expressions-regulieres/test.php?regex=2
```

Ces adresses sont un problème car le numéro "2" de cet exemple ne nous indique pas ce que nous donnera la page de destination. Fort heureusement, le nom du script et le paramètre GET portent chacun un nom explicite : nous pouvons ici raisonnablement deviner qu'il s'agit d'un test de l'expression régulière numéro 2 et que ce test s'inscrit dans le cadre d'un tutoriel en PHP dont je suis l'auteur.

Tout ceci a de l'importance pour Google.

Prenons un contre exemple : phpBB. C'est une application Web écrite en PHP et permettant de gérer un système de discussion entre utilisateurs. C'est système de forum parmi d'autres.

Cette application Web comprend un script nommé "viewtopic.php" qui accepte divers paramètres GET, parmi lesquels "t" et "p". Il est évident que ces noms ne signifient rien pour quelqu'un qui n'a pas étudié le fonctionnement de

l'application, ce qui n'est assurément pas le cas de Google. Ici, "p" signifie "post" (message) et "t" se rapporte à "topic" (sujet). Il n'est pas certain qu'un bot soit en mesure de deviner le contexte sémantique englobant ces deux lettres isolées, alors que c'était plus simple pour mon exemple sur les expressions régulières. Je ne me permettrai pas non plus d'affirmer que Google est en mesure de savoir que "viewtopic" est un condensé de deux mots distincts dont le plus important est "topic". Il aurait été préférable d'appeler le script d'une autre manière, par exemple "view-topic.php". En revanche, le paramètre "mode" peut prendre la valeur "edit", ce qu'un bot intelligent pourrait éventuellement interpréter comme : "c'est une page d'édition, donc *a priori* de moindre intérêt".

Il faut donc prendre garde à tous les éléments :

- Le nom du script **situe** le contexte ;
- Le nom d'un paramètre et sa valeur **précisent** le contexte.

Bien sûr, l'idéal pour un bot est de ne pas se compliquer la vie avec tous ces détails. Une URI sans paramètres est bien plus simple à gérer.

De plus, la présence d'un paramètre signifie que la page a potentiellement une quantité infinie de versions différentes. Un bot est comme vous et moi : ses capacités d'emmagasiner les informations sont limitées !

Enfin, l'ordre des paramètres dans l'URI ne change absolument pas le comportement du serveur. Les chaînes `viewtopic.php?p=1234&mode=edit` et `viewtopic.php?mode=edit&p=1234` sont différentes mais le serveur Web les traitera exactement de la même manière. Cependant, le bot devra effectuer un traitement supplémentaire pour ne pas traiter deux fois cette page s'il rencontre les deux URIs.

Tout cela suggère qu'il est préférable que le bot d'un moteur de recherche ne se rende pas compte qu'il est en train d'examiner une page dynamique. C'est ici qu'entre en jeu la technique de réécriture de liens décrite une dans l'un de mes cours (*cf.* les liens à la fin de ce tutoriel).

II-C - Les images :

Les images sont des éléments graphiques, visuels. Oui, c'est une évidence. Eh bien non, en réalité, ce n'en est pas une !


Les images ne sont pas intégrées au code source de la page HTML. Elles ne s'y trouvent que sous forme de balises HTML, rien de plus. Le bot n'est pas en mesure de les lire, de les analyser, d'en extraire le champ sémantique et de le comparer à celui du reste de la page. Pourtant, ce sont parfois des éléments fondamentaux de la page car ils permettent d'améliorer la lisibilité d'un long texte et d'expliquer simplement un concept complexe.

C'est pour cette raison qu'il ne faut pas négliger les images. Leurs balises permettent d'introduire des mots clefs que le bot ne manquera pas d'interpréter comme des mots particulièrement importants dans le contexte de la page. Il s'agit du contenu de la propriété *alt*, qui signifie "texte alternatif". À l'origine, cette propriété était uniquement destinée à proposer un élément visuel au cas où l'image ne serait pas disponible, afin de ne pas laisser le lecteur sur sa faim. Aujourd'hui, elle permet d'associer des mots clefs à une image ainsi que de renforcer le champ sémantique d'une page.

Exemple d'image avec un texte alternatif :

```

```

 **Hors sujet : n'utilisez pas la propriété *border* de vos images pour en supprimer la bordure lorsqu'elles se trouvent dans des liens ! Cela n'est pas le travail du HTML (structure du document) mais plutôt de votre CSS (présentation des éléments de votre document).**

Comment s'y prendre dans une CSS :

```
img
{
    border: 0px;
```

Comment s'y prendre dans une CSS :

```
}
```

II-D - Les tableaux : <table>

Beaucoup de webmestres ont tendance à utiliser l'élément <table> dans tout leur design. C'est une erreur ! En effet, la sémantique est incorrecte et, de plus, cet élément impose une structure visuelle, ce qui n'est pas l'objectif d'un document HTML mais de la feuille de style qui lui est appliquée (CSS). L'élément <table> est prévu pour présenter des données brutes telles qu'une liste de produits (par exemple "désignation + prix + quantité en stock"), non pour structurer toute une page Web.

À titre d'exemple, voici une utilisation tout à fait honorable d'un tableau : ma liste de courses d'hier...

N°	Quantité	Désignation	Coût
1	1	Mostassa Ant.	2,05
2	1	Tony Tom Q.P-3	1,20
3	2	S. Dindi/Poll.	3,20
4	2	Barrete. Patat.	1,20

II-E - Les divisions : <div>

Il n'est pas conseillé d'utiliser l'élément <table> pour structurer une page Web car il existe une meilleure solution.

Prenons comme exemple un texte composé de chapitres et de paragraphes. Une page simple permettant de mettre cela sur Internet est comme suit :

Page HTML mal structurée

```
<font size="3">Titre du texte</font><br /><br />
<font size="2">Chapitre 1</font><br />
Paragraphe 1.1<br /><br />
Paragraphe 1.2<br /><br />
Paragraphe 1.3<br /><br /><br />
<font size="2">Chapitre 2</font>
Paragraphe 2.1<br /><br />
Paragraphe 2.2
```

Bien sûr, ce HTML est complètement inefficace pour le référencement puisqu'aucune information de structure n'est fournie au bot, il ne peut donc trouver aucune sémantique. Ainsi, il nous faut diviser notre page en sections.

Je viens de vous donner le mot important : une **division**, à savoir l'élément <div> de votre code HTML.

Page HTML correctement structurée

```
<div>
  <h1>Titre du texte</h1>
</div>
<div>
  <div>
    <h2>Chapitre 1</h2>
    <p>Paragraphe 1.1</p>
    <p>Paragraphe 1.2</p>
  </div>
  <div>
    <h2>Chapitre 2</h2>
    <p>Paragraphe 2.1</p>
    <p>Paragraphe 2.2</p>
    <p>Paragraphe 2.3</p>
  </div>
</div>
```

Dans l'exemple ci-dessus, la structure du document est parfaitement explicite. Nous avons utilisé une division pour le titre puis une autre division pour le reste du texte. Ce texte est lui-même structuré en deux divisions, chacune composée d'un titre et de paragraphes. Il est possible de segmenter encore davantage, c'est selon le goût de chacun.

De plus, une organisation comme celle-ci facilite l'utilisation d'une CSS.

II-F - Les listes :

La liste à puces est un élément important de la page Web car elle permet d'identifier très rapidement ce qui constitue, devinez quoi, une liste d'éléments...

Elle peut être utilisée en tant que telle comme je l'ai fait à plusieurs reprises dans ce cours ou bien en combinaison avec une feuille de style pour construire un sommaire. Le plus important ici est d'avoir un document HTML correctement structuré.

Une technique très utile dans certaines situations est de construire une liste tout en cachant les puces au moyen d'une CSS. Cela vous donne un contrôle total sur l'apparence de votre liste, tout en conservant la sémantique du document. Cela vous permet d'utiliser des éléments graphiques sans que les anciens navigateurs (qui n'affichent pas ces éléments) soient trop lésés.

Ses avantages :

- Cohérence du code source HTML ;
- Permet aux navigateurs non graphiques de savoir qu'il s'agit d'une liste ;
- Permet aussi aux internautes de savoir qu'il s'agit d'une liste ;
- Possibilité de personnaliser l'apparence de la puce à l'aide de la CSS.

Exemple de liste dont on ne souhaite pas afficher les puces :

```
<ul style="list-style: none;">
  <li>Pseudo</li>
  <li>E-mail</li>
  <li>Site Web</li>
</ul>
```

II-G - Les formulaires : <form>

Grande est la tentation de gérer toutes les actions de l'utilisateur au moyen des liens (ancres HTML). Toutefois, cela risque de vous poser des problèmes...

La documentation officielle explique très clairement qu'une action de l'utilisateur doit être effectuée en mode POST exclusivement, tandis que le mode GET sert exclusivement à récupérer des informations pour l'utilisateur. Cela signifie que les liens ne peuvent servir qu'à récupérer des informations, pas à effectuer des actions. Pour cela, il faut utiliser un formulaire.

Si vous ne suivez pas cette norme, vous vous exposez à ce qu'un plugin du navigateur utilise la session de l'internaute pour suivre tous les liens du site. Bien sûr, puisque la session est utilisée, le plugin a accès à tous les liens qui ne sont affichés que dans la VIP de votre site et que vous ne destinez pas aux bots habituels. Je ne fais ici que relater une aventure qui est arrivée à des gens très connus (*cf.* les liens externes en fin d'article).

II-H - De manière générale

Structurez vos documents.


Utilisez les outils fournis par le HTML, ne négligez pas les balises. Donnez un titre à vos documents à l'aide de la balise <title> (de 60 à 80 caractères) et reformulez-le ou répétez-le dans votre balise <h1> (en vous permettant éventuellement une phrase plus longue).


Utilisez la sémantique du HTML :


- **em** (emphasis) : Attire l'attention sur un mot, ce qui se traduit habituellement par de l'italique ;
- **strong** : Indique un mot important (plus fort qu'*em*), ce qui se traduit habituellement par une mise en gras ;
- **cite** : Une citation ou une référence à d'autres sources d'information ;
- **q** (quote) : Dans le cas d'une citation, <cite> englobe l'auteur et <q> la citation elle-même ;
- **blockquote** : Une grande citation ;
- **dfn** (definition) : Le mot est défini à cet endroit de la page ;
- **code** : Code source, habituellement représenté par le navigateur à l'aide d'une fonte à chasse fixe ;
- **samp** (sample) : Exemple de résultat d'exécution de code source ;
- **var** : Une variable ;
- **abbr** : Une abréviation ;
- **acronym** : Un acronyme.

Certains de ces éléments (comme et) sont traduits visuellement dans certains navigateurs. Vous pouvez utiliser une feuille de style pour contrôler leur apparence.

Dans le cas d'<abbr> et <acronym>, il est possible de préciser deux attributs : "lang" et "title". Ils permettent de donner l'expression dans sa forme complète et d'en indiquer la langue.

 Apprenez à faire la différence entre certaines balises courtes comme d'une part et <i> et d'autre part leurs équivalents et . Les premiers sont des ordres donnés au navigateur : "je veux ça en gras (bold) et ça en italique" ; les autres sont des informations sémantiques : "ce mot est fort (strong), celui-ci est simplement à mettre en valeur (emphasis)".

 Certains éléments donnent l'impression de pouvoir être remplacés par des feuilles de style : nous reviendrons plus loin sur la distinction qu'il faut faire.

 N'oubliez pas d'inclure un "microsummary" (un titre alternatif et dynamique) dans chacune de vos pages. Le titre de votre page ne devrait pas être modifié par vos mises à jour, tandis que celui-ci doit l'être : cela informe les internautes de vos mises à jour. Les micro sommaires feront l'objet d'articles séparés : en attendant, vous trouverez tous les liens utiles en fin d'article. Firefox 2 permet d'utiliser les micro sommaires comme titre de ses bookmarks.

III - Utilisez les technologies à bon escient

III-A - Les images

Cela paraît trivial mais les images de vos documents doivent être aussi peu nombreuses que possible. D'une part, elles alourdissent le poids du document, ce qui génère un surplus d'utilisation de la bande passante du serveur et peut vous coûter une fortune dans certaines situations extrêmes. D'autre part, tel que je l'ai évoqué plus tôt, les bots de référencement ne peuvent pas interpréter le contenu des images avec la même facilité que le texte. Enfin, une image vous donne moins de contrôle à vous, webmestre, sur l'allure de votre page : modifier la CSS ne permet pas de modifier les images, donc pas tout le design de votre page.


III-B - Flash

Les documents Flash (extension .swf) sont très connus de tous les webmestres. Flash permet de créer des interfaces Web disposant d'animations qu'il serait extrêmement complexe de créer au moyen des technologies Web classiques comme JavaScript.

Ce n'est pas pour autant qu'il faut se laisser séduire. En effet, lors du référencement, les documents Flash sont encore moins analysés que les images, dans la mesure où leur balise HTML ne permet pas de leur attribuer de commentaire. Seul le nom du fichier peut renseigner sur leur contenu, or de nombreux webmestres négligent (trop) largement ce détail.

En conclusion, n'utilisez Flash que pour des animations isolées. Ne l'utilisez surtout pas pour faire le sommaire (le menu) de votre site et encore moins pour développer le site complet, car les bots seraient alors dans l'incapacité totale de visiter votre site au-delà de la page d'accueil !

III-C - AJAX



 **AJAX**, cette technologie qui a donné son nom à la tendance Web 2.0, est à utiliser avec parcimonie également. En effet, le contenu de la page étant modifié après son chargement, nous n'avons absolument aucune garantie que les crawlers seront en mesure de s'adapter. À vrai dire, il est tout à fait probable que ces crawlers n'interprètent pas du tout le langage JavaScript, ce qui leur occulte la totalité du contenu affiché au moyen d'AJAX.

Il est donc judicieux d'appliquer à votre utilisation d'AJAX des règles similaires à celles adoptées pour Flash : n'utilisez AJAX que pour ce qui est le plus utile, pour améliorer l'expérience utilisateur. Évitez de charger du contenu au moyen d'AJAX car les bots seront incapables de le référencer, ce qui serait bien dommage pour vous.

Concrètement, de bons exemples d'utilisation sont Google Maps et Google Suggest. Un mauvais exemple d'utilisation serait un site au contenu chargé dans des onglets dynamiques, c'est-à-dire remplis au moyen d'AJAX.

Réciproquement, ce n'est pas parce que vous ne référencez pas une certaine page qu'il faut y utiliser AJAX à tout prix.

III-D - Les feuilles de style

Qu'elles soient  **CSS**,  **XSL** ou d'une autre branche, les feuilles de style permettent de simplifier la mise en forme de votre site en vous permettant de définir une "classe" pour chacun de vos éléments HTML.

De nombreux webmestres utilisent et nomment ces classes de manière impropre. En effet, les classes CSS sont prévues pour être appliquées à un type d'élément HTML destiné à englober un contenu bien spécifique. Il peut s'agir d'une réponse dans un forum (regroupant le message, son titre, le nom de l'auteur, la date de rédaction *etc.*) ou bien simplement un message. En revanche, il ne doit pas s'agir ni du "menu de gauche" puisque l'emplacement dans la

page relève de la structure de votre document HTML, ni de la "case jaune", puisque cela relève des valeurs des propriétés de votre classe CSS (qui sont évidemment sujettes à modification).

Imaginez que vous ayez une classe "top-box" dans votre CSS : dès que votre document HTML sera modifié, par exemple lorsque vous décidez de déplacer l'en-tête vers l'un des bords de la page, vous devrez également modifier le nom de votre classe CSS ainsi que tous les documents HTML qui y font appel, ce qui vous conduira probablement à une erreur un jour ou l'autre. Évitez de donner un nom en rapport avec son emplacement dans la page ou avec la manière dont il apparaît. En revanche, je vous recommande vivement de choisir le nom en fonction du contenu des éléments auxquels la classe CSS doit être appliquée, puisque c'est en fonction de cela que vous remplissez les propriétés de cette classe. Des noms corrects sont "main-menu", "message-box", etc.

Par ailleurs, évitez d'utiliser vos feuilles de style pour cacher des éléments du contenu. En effet, cette tactique a toutes les chances d'être perçue comme une tentative d'ajout de mot clef à l'insu de l'internaute. Un débat fait rage quant à la capacité des crawlers de lire les feuilles de style : personnellement, je ne vous recommande pas de supposer qu'ils le sont. Partez du principe que les crawlers utilisent tous les éléments à leur disposition ; de toute manière, je ne doute pas qu'ils le feront un jour ou l'autre.

Enfin, utilisez les feuilles de style quand c'est nécessaire. Par exemple, mettre vos titres en caractères gras entre dans les attributions de CSS, tandis que mettre en gras un mot isolé de votre page devrait être fait au moyen de la balise HTML . Apprenez à faire la distinction entre "mettre en valeur" (ce qui comporte une sémantique et doit donc être fait au moyen d'une balise HTML) et simplement "mettre en forme" (ce qui doit être réservé aux feuilles de style afin d'éviter qu'une sémantique apparaisse où elle ne devrait pas).

III-E - Les headers, les redirections

Utilisez la redirection la plus efficace. Évitez les redirections en HTML ou en JavaScript, préférez un envoi de headers grâce à votre serveur Web ou à votre langage de script.

Pour de plus amples informations : [FAQ Comment faire une redirection ?](#)

De manière générale, il est recommandé de déterminer précisément quels sont les headers les plus adaptés et de les envoyer au client. De cette manière, vous indiquez au client toutes les informations dont il a besoin pour réagir de la manière la plus adaptée possible. Par exemple, dans le cas d'une page de déconnexion, le crawler peut recevoir un header UNAUTHORIZED et ainsi éviter de référencer inutilement cette page.

III-F - Contrôlez les liens sortants

N'importe quel webmestre est habitué à voir les utilisateurs de ses sites publier des liens vers d'autres sites. Sachez que la quantité de liens sortant de votre site est comptabilisée dans les paramètres de référencement. Plus vous avez un taux de liens élevé, moins vous paraissez sérieux (ou organisé, ce qui revient au même), ce qui fait baisser votre popularité. Pour réduire votre quantité de liens dans une situation où vos internautes peuvent publier ce qu'ils veulent, les moteurs de recherche se sont mis d'accord sur une propriété de la balise d'ancrage : rel="nofollow". Définissez une politique de filtrage de vos utilisateurs de confiance (par leur nombre de messages, leur ancienneté, ce que vous jugez nécessaire) et ajoutez cette propriété à tous les liens en lesquels vous ne pouvez pas avoir confiance : les moteurs de recherche n'en tiendront pas compte lors du référencement de vos pages.

Bien entendu, la réciproque est qu'il est totalement inutile de publier frénétiquement des liens vers votre site aux quatre coins du Web... Pour obtenir des backlinks, vous devez contacter le webmestre du site qui vous intéresse : d'une part, cette démarche vous oblige à faire un effort supplémentaire (courrier personnalisé = pas d'automatisme) ; d'autre part, cela donne le contrôle au webmestre hôte ; par conséquent, c'est une méthode fiable.

Utilisation de la propriété rel='nofollow' :

```
<a href="http://www.lien-publicite.com/" title="Publicité" rel="nofollow">Site de publicité</a>
```

IV - Quelques outils

IV-A - Pourquoi utiliser des outils ?

Ne croyez pas que mettre un site en ligne soit suffisant pour qu'il fonctionne... Non, en plus de le faire vivre, il faut le surveiller afin d'effectuer des ajustements techniques au quotidien et afin de modérer ce que vos utilisateurs publient. Il ne s'agit pas de censurer mais plutôt d'éviter les éclats : à vous de trouver les mots pour justifier vos actions.

En bref, vous ne pouvez tout simplement pas vous permettre de surveiller la technique de tout le site en plus de son contenu. À cause de votre condition d'humain, vous n'en avez pas les capacités : vous avez besoin d'outils pour y parvenir convenablement.

IV-B - Les statistiques

Les outils de statistiques (dont je vous laisse le soin de faire le tri) sont un merveilleux moyen de connaître vos utilisateurs, leur parcours dans votre site, leurs habitudes, le succès de chaque page de votre site, *etc.* Tant que possible, cartographiez votre site. Repérez les zones d'ombre ainsi que celles qui ont beaucoup de succès, déterminez la cause des écarts. Ils peuvent également vous permettre de savoir quels mots clefs sont les plus utiles pour faire venir les visiteurs.

Tout cela vous permet d'apporter des solutions à vos problèmes.

Pour obtenir de meilleurs résultats, étudiez également les "referers" pour connaître l'origine de vos visiteurs. Étudiez alors le site en question, apprenez tout ce que vous pouvez dessus, analysez sa méthode de référencement puis, si cela en vaut la peine, prenez contact pour mettre en place un partenariat.

IV-C - La publicité

Votre site gagnera à se faire connaître. Bien sûr, nous avons déjà parlé de méthodes permettant d'éviter le spam de liens... N'essayez pas de faire votre publicité en tant qu'utilisateur des autres sites, ce serait mal vu : comme suggéré ci-dessus, vous gagnerez à contacter directement le webmestre. Cela vous permettra de mettre en place des programmes d'échanges de liens et de bénéficier de la popularité du site partenaire. Toutefois, prenez garde à ne pas effectuer ce genre de manipulation avec un site concurrent, vous y perdriez tous les deux : il est préférable de trouver des sites complémentaires.

Plutôt que de faire connaître votre site vous-même, ce qui ne donne que rarement des résultats satisfaisants, il est préférable de compter sur vos utilisateurs : ils sont plus nombreux que vous, ce qui signifie qu'ils ont un potentiel publicitaire plus important que vous seul. C'est principalement pour cela qu'il vous faut veiller au **contenu** de votre site : pour un bon site, la publicité se fait d'elle-même !

Enfin, ne négligez pas les diverses solutions de publicité. Vous avez la possibilité d'apparaître dans des programmes comme AdWords (Google) ou bien de négocier des campagnes de pub personnalisées avec certains sites (par exemple ce que nous avons sur Developpez.com). N'hésitez pas non plus à mettre en place votre propre programme de publicité sur votre site : bien utilisé, ce système peut vous permettre d'attirer des visiteurs.

Cela dit, soyez vigilant avec la publicité. De nombreux internautes sont rebutés par les bannières trop agressives, il vous faudra donc veiller à ne pas les effrayer. À votre charge de trouver la bonne mesure publicité / discrétion.

V - Conclusion






V-A - Épilogue

Vous aurez compris que les crawlers préfèrent parcourir les documents lorsqu'ils sont au format texte. Utilisez la sémantique que vous offrent les balises HTML pour structurer vos documents. Donnez des indications en tirant parti des attributs des balises HTML ainsi que des noms des fichiers, des paramètres GET et des classes CSS.




En suivant ces conseils simples, votre site montera peu à peu en popularité. Il ne vous reste plus qu'à vous concentrer sur le contenu de votre site, qui reste indéniablement le paramètre le plus important de l'équation.

V-B - Liens

Ressources de Developpez

-  [Un exemple de liste à puces couplée à une CSS](#), par Pierre-Baptiste Naigeon ;
-  [Quels sont les headers HTTP ?](#)
-  [Comment récupérer tous les liens d'une page Web](#), par Guillaume Rossolini ;
-  [Comment créer un formulaire HTML](#), par Guillaume Rossolini ;
-  [Comment utiliser les titres en \(X\)HTML](#), par Alen Grakalic ;
- [FAQ La FAQ Webmarketing](#).

Ressources externes

- [37signals.com](#) :  [L'histoire de 37signals et de Google Web Accelerator "le cliqueur fou"](#) ;
- [Mozilla.org](#) :  [Définition du microsommaire](#) ;
- [Mozilla.org](#) :  [Comment générer un microsommaire](#).

V-C - L'auteur

 *Guillaume Rossolini est développeur et formateur pour sa société **Alveod**.*

1 : L'absence de la lettre "e" n'est pas la seule contrainte imposée par Georges Perec pour son roman lipogrammatique. Les critiques sont d'avis que c'est l'ensemble de ces contraintes qui font la véritable prouesse.

2 : Raymond Queneau a participé à la fondation d'un mouvement littéraire du nom d'Oulipo ("ouvroir de littérature populaire", dont était membre Georges Perec). Ce mouvement a pour objectifs de développer des techniques d'écriture alternative et de rechercher les auteurs ayant utilisé ces techniques avant la création officielle du mouvement. Peut-être connaissez-vous l'écriture automatique ou le cadavre exquis ? Loin d'être expert en la matière, j'ai lu Les Fleurs Bleues sans me rendre compte un seul instant du fait que la structure repose entièrement sur les mathématiques. Comme quoi, il est * parfois * intéressant de sortir des cadres pré établis...