

## HTML pour les nuls et les impatientes (v1.0)

### Introduction

Le langage HTML est très simple, il suffit de quelques minutes pour en saisir le principe et quelques heures pour commencer à l'utiliser.

#### Objectifs de l'article

Cet article s'adresse à toute personne désirant optimiser un peu la rédaction d'articles ou de pages dans WordPress, ou de soigner la présentation de ses commentaires dans les blogs pour les rendre plus attractifs.

#### Les limites de l'article

Cet article n'est pas un tutoriel HTML, beaucoup de notions importantes sont laissées de côté. Il ne vous permettra pas, à fortiori, de vous lancer dans la création d'un site Web. Il existe sur le net quantité d'excellents sites permettant l'apprentissage en profondeur du HTML<sup>1</sup>.

Il m'arrive souvent, tout au long de l'article, de donner en exemple des instructions fausses. Je traduis par exemple « display:none; » par « affichage : ne pas afficher ». Mes instructions de code sont pédagogiques et ne doivent pas être prises au pied de la lettre.

### HTML Première approche

#### À quoi sert le HTML ?

Le HTML est le langage utilisé par votre navigateur (Firefox, Chrome, Internet Explorer, Safari...) pour afficher sur votre écran le contenu d'un document.

Pour ce présent texte, par exemple, le navigateur a besoin de savoir ce qu'il doit afficher en titre, en sous titre, en texte ordinaire, en image...

Sans HTML, imaginez le début de cet article, cela donnerait quelque chose comme :

```
HTML pour les nuls et les impatientes Introduction Le langage HTML est très simple, il suffit de quelques minutes pour en saisir le principe et quelques heures pour commencer à l'utiliser. Objectifs de l'article Cet article s'adresse à toute personne désirant optimiser un peu la rédaction d'articles ou de pages dans WordPress, ou de soigner la présentation de ses commentaires dans les blogs pour les rendre plus attractifs. Les limites de l'article Cet article n'est pas un tutoriel HTML, beaucoup de notions importantes sont laissées de côté et il existe sur le net quantité d'excellents sites permettant l'apprentissage du HTML. À quoi sert le HTML ?
```

1 Voir quelques liens en fin de document

Le HTML est le langage utilisé par votre navigateur (Firefox, Chrome, Internet Explorer, Safari...) pour afficher sur votre écran le contenu d'un document. Pour ce présent texte, le navigateur a besoin de savoir ce qu'il doit afficher en titre, en sous titre, en texte ordinaire, en image...

## Fonctionnement imagé du HTML :

Très sommairement, le HTML va fonctionner ainsi : (voir tableau ci-dessous)

On distingue les éléments de structure (à gauche) et les éléments de contenu - le fond (à droite).

Nous allons donc reprendre les premiers mots de ce présent article en séparant le contenu en plusieurs blocs auxquels seront accolés des informations de structure et de mise en forme.

### Document HTML

STRUCTURE	CONTENU
<b>Titre principal</b> Saut de ligne avant et après, Taille de police très grosse, couleur noire, police avant garde, texte centré,	HTML pour les nuls et les impatientes
<b>Titre de niveau 2</b> Saut de ligne avant et après, Taille de police assez grosse, couleur noire, police Helvetica, texte aligné à gauche,	Introduction
<b>Paragraphe</b> Saut de ligne avant et après, Police Times, de taille 10, Texte justifié	Le langage HTML est très simple, il suffit de quelques minutes pour en saisir le principe et quelques heures pour commencer à l'utiliser.
<b>Titre de niveau 2</b> Saut de ligne avant et après, Taille de police assez grosse, couleur noire, police Helvetica, texte aligné à gauche,	Objectifs de l'article
<b>Paragraphe</b> Saut de ligne avant et après, Police Times, de taille 10, Texte justifié	Cet article s'adresse à toute personne désirant optimiser un peu la rédaction d'articles ou de pages dans WordPress, ou de soigner la présentation de ses commentaires dans les blogs pour les rendre plus attractifs.
<b>Titre de niveau 2</b> Saut de ligne avant et après, Taille de police assez grosse, couleur noire, police helvetica, texte aligné à gauche,	Les limites de l'article

Le navigateur va afficher chaque bloc de contenu en ajoutant les enrichissements (forme) indiqués en regard.

## Un mot sur les CSS :

Un document CSS (feuille de styles) va simplifier le travail de l'humain rédacteur de l'article et du logiciel de navigation en regroupant toutes les instructions de mise en forme à un seul endroit.

Reprenons notre tableau ci-dessus : Les informations de structure vont être regroupées (factorisées

diront certains) et stockées à proximité.

Nous allons donc créer un document CSS qui contiendra par exemple :

## Feuille de style CSS

<b>Titre principal</b>	Saut de ligne avant et après, Taille de police très grosse, couleur noire, police avant garde, texte centré,
<b>Titre de niveau 2</b>	Saut de ligne avant et après, Taille de police assez grosse, couleur noire, police Helvetica, texte aligné à gauche,
<b>Paragraphe</b>	Saut de ligne avant et après, Police Times, de taille 10, Texte justifié

Notre document HTML deviendra donc

## Document HTML

STRUCTURE	CONTENU
<b>Titre principal</b> (voir feuille de style CSS)	HTML pour les nuls et les impatientes
<b>Titre de niveau 2</b> (voir feuille de style CSS)	Introduction
<b>Paragraphe</b> (voir feuille de style CSS)	Introduction Le langage HTML est très simple, il suffit de quelques minutes pour en saisir le principe et quelques heures pour commencer à l'utiliser.
<b>Titre de niveau 2</b> (voir feuille de style CSS)	Objectifs de l'article
<b>Paragraphe</b> (voir feuille de style CSS)	Cet article s'adresse à toute personne désirant optimiser un peu la rédaction d'articles ou de pages dans WordPress, ou de soigner la présentation de ses commentaires dans les blogs pour les rendre plus attractifs.
<b>Titre de niveau 2</b> (voir feuille de style CSS)	Les limites de l'article

Autre avantage : Si vous avez un document assez long et que vous voulez changer la couleur de tous les sous-titres de niveau 2, il suffira de modifier un seul élément dans la feuille de style CSS.

## Langage à balises

Le HTML est un langage à balises. Chaque élément de contenu concerné par une structure ou par une mise en forme particulière sera délimité par une balise de début et une balise de fin, que l'on nomme respectivement « balise ouvrante » et « balise fermante »

Les balises se reconnaissent par les signes < et >

Les balises fermantes sont quasi-identiques aux balises ouvrantes, avec un </ en début.

Exemples : `<strong>`Bloc affiché en caractères gras`</strong>`

`<em>`Bloc affiché en italiques`</em>`

`<p>`Bloc de paragraphe`</p>`

`<h>`Bloc de titre`</h>`

## Attention à la différence avec les shortcodes

Les « shortcodes » que l'on trouve souvent dans les blogs ne sont pas des balises html, mais des raccourcis de programmation qui peuvent parfois, mais pas toujours, avoir des résultats équivalents. Ils sont délimités par des crochets [ et ].

Exemples assez fréquent : `[b]xxxx[/b]` est équivalent à `<strong> xxxx</strong>`

`[i]xxxx[/i]` est équivalent à `<em>xxxx</em>`

## Arguments dans la balise

Outre son titre, la balise peut contenir des « arguments » (instructions complémentaires). La lecture de ces arguments est parfois assez déroutante et nous allons nous y attarder un peu.

Il ne suffit pas toujours d'indiquer que tel bloc de contenu appartient à un certain type d'élément, un lien doit avoir une adresse, une image doit avoir un lieu de stockage... Prenons le cas d'un lien hypertexte (la balise des liens est `<a>`)

### **Exemple :**

Utilisons le code HTML ci-dessous :

```
Pour en savoir plus, <a>suivez ce lien</a>
```

Le navigateur affichera ce texte :

Pour en savoir plus, [suivez ce lien](#)

Cela semble correct, les mots « suivez ce lien » s'affichent bien dans une forme standard indiquant la présence d'un lien hypertexte (coloration et soulignement), mais si on clique sur le lien, il ne se passe rien.

Il va falloir ajouter une information de destination dans la balise `<a>`, cette information supplémentaire s'appelle un argument et est de la forme « `href="destination"` »

Le code HTML devra donc être :

```
Pour en savoir plus, <a href="http://www.google.fr" >suivez ce lien</a>
```

Les arguments se mettent dans la balise ouvrante, les uns à la suite des autres, séparés par une espace.

Dans l'exemple ci-dessous, on indique en plus de l'argument de destination un argument cible (`target`) qui ordonne au navigateur d'ouvrir le lien dans une autre nouvelle fenêtre.

```
Pour en savoir plus, <a href="http://www.google.fr" target=" blank" >suivez ce lien</a>
```

## L'argument de mise en forme : `class="..."`

Un argument très fréquent va indiquer au navigateur d'aller chercher la mise en forme du contenu dans une feuille de style, il s'agit de l'argument « `class="xxxxx"` »

## Exemple :

Dans la feuille de style se trouve la classe « .red » définie par

```
.red { couleur de police : rouge ; soulignement : aucun ; épaisseur : gras ; }
```

Le code HTML

```
Pour en savoir plus, <a class="red" href="http://www.google.fr" > suivez ce lien</a>
```

provoquera l'affichage suivant :

Pour en savoir plus, **suivez ce lien**

## L'argument de mise en forme : id="..."

Cet argument est similaire à l'appel à une classe, mais il référence un bloc unique désigné par son nom, alors que l'argument class="..." peut s'appliquer à de nombreux blocs.

## L'argument de mise en forme : style="..."

Cet argument est aussi très fréquent, il est utilisable dans toutes les balises et permet de rentrer directement du code CSS de mise en forme. Ce n'est pas une solution élégante car on ne délègue plus la mise en forme au fichier CSS, mais on l'utilise souvent à titre exceptionnel ou provisoire, pour étudier en direct le fonctionnement d'un style. Si ce style doit être réutilisé ailleurs dans le document, la bonne pratique est de l'intégrer dans une classe CSS.

## Lire un code source

Pour lire un code HTML, il suffit donc le plus souvent d'identifier le type de balise par son titre et de se rendre directement au signe de fin de balise ( > ) en ne lisant les arguments que si on en a besoin.

À noter que certains éléments HTML n'ont pas de balise fermante. Il s'agit principalement des images et des sauts de ligne. Nous y reviendrons.

## Imbrication des balises

Les balises HTML peuvent s'imbriquer les unes dans les autres, mais elles ne doivent pas se croiser. On ne peut pas mettre en gras avec la balise <strong> la fin d'un paragraphe et le début du suivant. La balise <strong> du premier paragraphe doit se fermer avant la fin du premier paragraphe et il faut placer une nouvelle balise <strong> dès le début du deuxième paragraphe.

Ceci est interdit (<p> délimite les paragraphes)

```
<p>Texte divers<strong>Texte en gras</p>
<p>Texte en gras</strong>Texte divers</p>
```

Ceci est obligatoire

```
<p>Texte divers<strong>Texte en gras</strong></p>
<p><strong>Texte en gras</strong>Texte divers</p>
```

# Structure d'un document HTML

---

Un document HTML est encadré par les balises `<html>`

À l'intérieur d'un document HTML, on trouve deux zones importantes, la zone d'en-tête (`<head>`) qui comporte des indications techniques destinées au navigateur et la zone de contenu proprement dite (`<body>`). Cela se représente comme suit :

```
<html>
  <head>
    Zone technique d'en-tête de document
  </head>
  <body>
    Contenu du document
  </body>
</html>
```

À l'intérieur du « body » (entre les balises `<body>` et `</body>`) nous allons trouver diverses balises, voyons en l'essentiel.

## Les balises

---

### Titres, paragraphes et sauts

#### Titres

Les titres sont des balises `<h>`, nommées de `<h1>` à `<h6>` selon leur niveau.

Le plus souvent, le titre de la page est un titre de niveau 1 (`<h1>`)

On réserve donc au contenu proprement dit les titres `<h2>`, `<h3>`, `<h4>`.

Bien que cela soit possible, on ne va que rarement jusqu'aux titre de niveaux 5 et 6.

#### Paragraphes

Les paragraphes sont encadrés par les balises `<p>`

#### Note pour WordPress :

*L'éditeur texte de WordPress gère automatiquement quelques balises, dont la balise `<p>`. C'est assez déroutant, les balises semblent disparaître d'elles mêmes. (Faut s'habituer !)*

#### Sauts

La balise `<br>` provoque un retour ligne et la balise `<hr>` trace un trait séparateur horizontal.

Ces deux balises ne peuvent pas avoir de contenu et au lieu d'écrire `<br></br>` ou `<hr></hr>`, on combine les balises ouvrantes et fermantes en une seule, en écrivant `<br />` et `<hr />`

(Ce sera la même chose pour les images)

Une astuce d'aération des pages consiste à utiliser la balise `<hr />` avec un réglage CSS qui rendra le trait invisible. Pour cela, dans le fichier css, on crée une classe nommée par exemple « peterBrady »

```
.peterBrady {affichage : ne pas afficher;}
```

et dans le document HTML, on écrira :

```
<hr class="peterBrady" />
```

À ce niveau de connaissance, nous pouvons imaginer un document HTML comme ci-dessous :

```
<html>
  <head>
    Zone technique d'en-tête de document
  </head>
  <body>
    <h1>Titre de la page</h1>
    <h2>Sous titre</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,</p>
    <h3>Sous-sous-titre</h3>
    <p>Lorem ipsum dolor sit amet, consectetur <br /> adipiscing elit. Sed non risus. Suspendisse <br /> lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,</p>
    <h3>Sous-sous-titre</h3>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,</p>
    <h2>Sous-titre</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,</p>
  </body>
</html>
```

## La mise en forme (enrichissements)

Il ne faut pas abuser des enrichissements qui peuvent conduire à un aspect brouillon. Néanmoins il peut être parfois utile et recommandé de mettre un mot en caractères **gras** ou en *italiques*.

Les caractères gras sont délimités par la balise `<strong>` et les italiques par la balise `<em>`<sup>2</sup>.

Ces balises viennent donc s'imbriquer à l'intérieur des balises `<p>` (ou plus rarement `<h>`).

Notre document peut maintenant devenir :

```
<html>
  <head>
    Zone technique d'en-tête de document
  </head>
  <body>
    <h1>Titre de la page</h1>
    <h2>Sous titre</h2>
    <p>Lorem ipsum dolor sit amet, <strong>consectetur elit</strong>. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,</p>
    <h3>Sous-sous-titre</h3>
    <p>Lorem <em>ipsum dolor sit amet</em>, consectetur <br />
```

<sup>2</sup>Notons qu'on trouve parfois les balises `<b>` et `<i>` qui sont des restes d'anciennes versions du langage HTML.

```
adipiscing elit. Sed non risus. Suspendisse <br />
lectus tortor, dignissim sit amet, adipiscing nec, ultricies
sed,</p>
<h3>Sous-sous-titre</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed,</p>
<h2>Sous-titre</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec,
ultricies sed,</p>
</body>
</html>
```

## Les blocs

### La balise <div>

À l'image d'un livre qui peut contenir plusieurs sections – Préface, Table des matières, bibliographie, contenu principal etc. – obéissant à des règles de typographie spécifiques, on peut séparer un document HTML en diverses zones par les balises <div>.

On pourra dans l'idéal avoir des réglages différents pour l'en-tête affiché, contenant par exemple le menu du site et le titre de la page, souvent désigné « Header », pour le corps du document proprement dit, souvent désigné « Container » et pour le bas de page, souvent désigné « Footer ».

Chaque zone pourra avoir ses propres réglages CSS et une balise <p> ne réagira alors pas de la même façon si elles se trouve dans l'une ou l'autre de ces sections.

Les zones <div> sont souvent caractérisées par des arguments de type « id="nom" » désignant un nom unique, « class="xxx" » ou « style="xxx" » désignant des styles CSS particuliers.

### La balise <span>

<span> permet de faire des divisions à l'intérieur d'un bloc ou d'un paragraphe, sans provoquer de retour à la ligne.

Exemple : le code HTML ci-dessous

```
<p>Un exemple de texte <span class="doubleEspace">espacé</span> sans retour
ligne.</p>
```

donnerait

Un exemple de texte E S P A C É sans retour ligne.

Bien sûr, nous aurions préalablement défini en CSS la classe « .doubleEspace »<sup>3</sup>

```
.doubleEspace {Casse : Majuscule ; espacement : double ;}
```

### La balise <blockquote>

<blockquote> sert à mettre un texte en exergue. À utiliser avec parcimonie, elle permet de mettre en valeur une citation, un petit texte important.

---

3 Rappel : le code indiqué est inexact, il n'est qu'une illustration pédagogique.

## Exemple :

Après avoir défini en CSS la classe .exergue

```
.exergue { retrait de ligne : 7cm ; largeur de texte : 6cm ; couleur de  
texte : gris ; style de texte : italiques ; taille de police : 24 ;  
Guillemets de départ : Guttman ; interligne : réduit ;}
```

Le code HTML ci-dessous

```
<blockquote>C'est en lisant qu'on devient liseron</blockquote>
```

donnerait

“ *C'est en lisant qu'on  
devient liseron.* ”

Nous l'avons dit plus haut, toutes ces balises peuvent s'imbriquer, comme ci-dessous la balise <div> de navigation à l'intérieur de la balise <div> de Header.

Notre document HTML :

```
<html>  
  <head>  
    Zone technique d'en-tête de document  
  </head>  
  <body>  
    <div id= "header">  
      <div id="navigation">Le menu</div>  
      <h1>Titre de la page</h1>  
    </div>  
    <div id="container">  
      <blockquote>Le chapeau de l'article</blockquote>  
      <h2>Sous titre</h2>  
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed  
      non risus. Suspendisse lectus tortor, dignissim sit amet,  
      adipiscing nec, ultricies sed,</p>  
      <h3>Sous-sous-titre</h3>  
      <p>Lorem ipsum dolor sit amet, consectetur <br />  
      adipiscing elit. Sed non risus. Suspendisse <br />  
      lectus tortor, dignissim sit amet, adipiscing nec, ultricies  
      sed,</p>  
      <h3>Sous-sous-titre</h3>  
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
      Sed non risus. Suspendisse lectus tortor, dignissim sit  
      amet, adipiscing nec, ultricies sed,</p>  
    <h2>Sous-titre</h2>  
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed  
      non risus. Suspendisse lectus tortor, dignissim sit amet,  
      adipiscing nec, ultricies sed,</p>  
    </div>  
    <div id="footer">  
      Informations de pied de page, copyright...  
    </div>  
  </body>  
</html>
```

## Les images

Les images sont introduites par la balise `<img />`. On remarque le « / » juste avant le signe `>` qui s'explique, comme pour les balises `<hr />` et `<br />` vues précédemment, par le fait qu'une image est un objet à part entière et n'a donc pas de fin.

La balise `<img />` nécessite plusieurs arguments, dont au moins l'adresse du fichier image sur le serveur (ou sur le web). Cet argument est de la forme « `src="adresse"` »

```

```

Ensuite, il est de bon ton de nommer l'image (argument « `title` »), de prévoir un texte de remplacement pour le cas où elle ne s'afficherait pas (argument « `alt` ») et de préciser une éventuelle légende (argument « `caption` ») et tout un tas d'options d'affichage.

Une image standard se code ainsi :

```

```

## Les liens

Comme nous l'avons vu au début de ce document, les liens hypertexte sont délimités par les balises `<a>` avec un argument de destination (`href`), très souvent un argument de cible (`target`), parfois un argument de titre (`title`) et d'autres encore.

Un lien standard se code ainsi :

```
<a href="url de destination">Texte du lien</a>
```

## Caractères accentués

On trouve parfois des caractères étranges du genre `&nbsp;` ou `&eacute;`;

Ces signes viennent d'une époque où les caractères codés sur 7 bits étaient limités en nombre et ne permettaient pas d'écrire des ç, æ, et autres Â ou ù...

On utilisait alors un code HTML particulier, commençant par « `&` » et finissant par « `;` »

Le joli mot « théâtre » se transforme en « `th&eacute;âtre` »

Le plus simple sera de se conformer à une table de conversion sur Internet.

## Aller plus loin :

Dans ce document, nous n'avons pas abordé certaines balises courantes : les listes, les tableaux et quantité d'autres balises moins fréquentes.

Pour ce qui est des tableaux, leur usage est maintenant assez limité. Il fut un temps où la structure des pages faisait appel à un emploi massif des tableaux. Cela se reconnaît dans certains codes sources par l'accumulation de balises `<tr>` et `<td>`.

Heureusement, cet usage est révolu et on réserve les tableaux à l'affichage de données tabulaires, des tarifs, des horaires, des statistiques...

Pour écrire du HTML, ne prenez jamais exemple sur un site contenant des balises `<tr>` et `<td>` ou contenant des balises en majuscules.

## Où trouver de l'aide ?

---

On trouve quantité de ressources sur le net, vous n'aurez que l'embarras du choix.

Veillez cependant à regarder la date de parution de l'article. Vérifiez aussi que les balises présentées soient bien en minuscules. Dans le cas contraire, le HTML concerné sera trop ancien et vous risquerez de vous égarer sur des balises aujourd'hui inutilisées, comme la balise <FONT>.

Beaucoup de sites vont vous proposer d'apprendre à créer un site Web, ce n'est peut-être pas votre intention, il vous faudra donc filtrer les informations.

### Quelques liens :

<http://www.startyourdev.com/html/tag-html-index>

<http://www.siteduzero.com/informatique/tutoriels/apprenez-a-creer-votre-site-web-avec-html5-et-css3>

<http://www.alsacreations.com/tutoriels/>