

# PROGRAMME WEB 1

- I. HTML**
- II. FEUILLE DE STYLE OU CSS**
- IV. PHP 5**
- V. INTERACTION AVEC BASE DE DONNÉES : REQUÊTE / MYSQL / APACHE**
- VI. LES CMS (CONTENT MANAGER SYSTEM)**
- VII. PROJET**

## Chapitre1 : fonctionnement des sites web

### Introduction

Les concepts du Web (*toile en anglais*), ayant comme acronyme le www (World Wide Web), a été mis au point en 1991 au CERN (*Centre Européen de Recherche Nucléaire*) par une équipe de chercheurs à laquelle appartenaient Tim Berners-LEE, le créateur du concept d'hyperlien, considéré aujourd'hui comme le père fondateur du Web qui repose sur l'utilisation d'hyperliens pour naviguer entre des documents (pages web) grâce à un logiciel appelé navigateur (browser en anglais).

Une page web est ainsi un simple fichier texte écrit dans un langage de description appelé *HTML*, permettant de décrire la mise en page du document et d'inclure des éléments graphiques ou bien des liens vers d'autres documents à l'aide de balises.

Le protocole HTTP qui permet cette communication entre les pages web et permet de lier des documents hébergés par des ordinateurs distants (**serveurs web**). Ce pendant les ordinateurs qui accède à ces pages des serveurs web sont appelées client web et représenté par le navigateur. Sur

Internet les documents sont ainsi repérés par une adresse unique, appelée URL, permettant de localiser une ressource sur n'importe quel serveur du réseau internet.

Vous lancez un programme appelé le navigateur web en cliquant sur l'une de ces icônes par exemple:



Avec le navigateur, vous pouvez consulter n'importe quel site web. Voici par exemple un navigateur affichant le célèbre site web Wikipédia :



## Comment créer des sites web comme celui-ci ?

Tous les sites web sont basés sur des langages informatiques, ils sont incontournables et universels aujourd'hui. Ils sont à la base même du Web. Le langage HTML a été inventé le premier par Tim Berners-Lee en 1991. Tim Berners-Lee suit encore aujourd'hui avec attention l'évolution du Web. Il a créé le World Wide Web Consortium (W3C) qui définit les nouvelles versions des langages. Il a par ailleurs créé plus récemment la World Wide Web Foundation, qui analyse et suit l'évolution du Web.



*Tim Berners-Lee, inventeur du Web*

**A**ttention il ne faut pas confondre Internet et le Web. Internet est donc un grand ensemble de réseaux qui comprend, entre autres : le Web, les e-mails, la messagerie instantanée, etc. *Tim Berners-Lee* n'est donc pas l'inventeur d'Internet, c'est "seulement" l'inventeur du Web.

Les langages HTML et CSS sont à la base du fonctionnement de tous les sites web. Quand vous visitez un site avec votre navigateur, il faut savoir que derrière, des rouages s'activent pour permettre au site web de s'afficher. L'ordinateur se base sur ce qu'on lui a expliqué en HTML et CSS pour savoir ce qu'il doit afficher.

HTML et CSS sont deux "langues" qu'il faut savoir parler pour créer des sites web. C'est le navigateur web qui fera la traduction entre ces langages informatiques et ce que vous verrez s'afficher à l'écran. Pour voir le code HTML et CSS de la page vous pouvez faire :

*Clic droit sur la page ➔ afficher code source du menu contextuel*

## **HTML et CSS : deux langages pour créer un site web**

Il faut indiquer où placer le texte, insérer des images, faire des liens entre les pages, etc. pour créer un site web. On doit donc donner des instructions à l'ordinateur, donc Il ne suffit pas simplement de taper le texte qu'il y aura dans son site comme on le ferait dans un traitement de texte.

## Le rôle de HTML et CSS

Ces deux langages se *complètent*:

- **HTML** (*HyperText Markup Language*) : il a fait son apparition dès 1991 lors du lancement du Web et la version actuelle est le HTML5. Son rôle est de gérer et organiser le contenu. C'est donc en HTML que vous écrirez ce que vous souhaitez que la page affiche : du texte, des liens, des images... Vous direz par exemple : "Ceci est mon titre, ceci est mon menu, voici le texte principal de la page, voici une image à afficher, etc..".
- **CSS** (*Cascading Style Sheets*, aussi appelées *Feuilles de style*) : le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleur, taille du texte...). Ce langage est venu compléter le HTML en 1996.

Vous pouvez très bien créer un site web uniquement en HTML, mais celui-ci ne sera pas très beau : l'information apparaîtra "brute". C'est pour cela que le langage CSS vient toujours le compléter. Comme vous vous en doutez, le CSS a besoin d'une page HTML pour fonctionner.

### Les différentes versions de HTML

- HTML 1 : c'est la toute première version créée par Tim Berners-Lee en 1991.
- HTML 2 : la deuxième version du HTML apparaît en 1994 et se finira en 1996 avec l'apparition du HTML 3.0. C'est cette version qui posera en fait les bases des prochaines versions du HTML. Les règles et le fonctionnement de cette version sont donnés par le W3C (tandis que la première version a été créée par un seul homme).
- HTML 3 : apparue en 1996, cette nouvelle version du HTML rajoute de nombreuses possibilités au langage comme les tableaux, les applets, les scripts, le positionnement du texte autour des images etc...
- HTML 4 : il s'agit de la version la plus répandue de HTML (plus précisément il s'agit de HTML 4.01). Elle apparaît pour la première fois en 1998, et propose l'utilisation de frames (qui découpent une page web en plusieurs parties), des tableaux plus complexes, des améliorations sur les formulaires etc... Mais surtout, cette version permet pour la première fois l'utilisation de feuilles de style CSS

- HTML 5 : c'est la dernière version. Encore assez peu répandue, elle fait beaucoup parler d'elle car elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc.

### Les différentes versions de CSS

- CSS 1 : dès 1996, la première version du CSS est utilisable. Elle pose les bases de ce langage qui permet de présenter sa page web, comme les couleurs, les marges, les polices de caractères etc...
- CSS 2 : apparue en 1999 puis complétée ensuite par CSS 2.1, cette nouvelle version de CSS rajoute de nombreuses options. On peut désormais utiliser des techniques de positionnement très précises qui nous permettent d'afficher des éléments où on le souhaite sur la page.
- CSS 3 : c'est la dernière version, qui apporte des fonctionnalités particulièrement attendues comme les bordures arrondies, les dégradés, les ombres, etc.

### Quel logiciel je vais avoir besoin pour créer mon site web ?

Il existe effectivement de nombreux logiciels dédiés à la création de sites web. On peut classer ces logiciels de création de site web en deux catégories :

- Les WYSIWYG (What You See Is What You Get - Ce Que Vous Voyez Est Ce Que Vous Obtenez) : ce sont des programmes qui se veulent très faciles d'emploi, ils permettent de créer des sites web sans apprendre de langage particulier. Parmi les plus connus d'entre eux : *Nvu*, *Microsoft Expression Web*, *Dreamweaver*... Leur principal défaut est la qualité du code HTML et CSS qui est automatiquement généré par ces outils, souvent d'assez mauvaise qualité. Un bon créateur de site web doit tôt ou tard connaître HTML et CSS.
- Les éditeurs de texte : ce sont des programmes dédiés à l'écriture de code. On peut en général les utiliser pour de multiples langages, pas seulement HTML et CSS. Ils se révèlent être de puissants alliés pour les créateurs de sites web. Si Notepad++ ne vous convient pas, vous pouvez essayer : jEdit, PSpad, ConTEXT...

Il est conseillé d'installer plusieurs navigateurs sur son ordinateur pour s'assurer que son site fonctionne correctement sur chacun d'eux. Au moins sur les plus utilisés comme Google Chrome, Mozilla Firefox et Internet Explorer. Cela est dû au fait que les navigateurs ne connaissent pas toujours les dernières fonctionnalités de HTML et CSS. Par exemple, Internet Explorer a longtemps été en retard sur certaines fonctionnalités CSS. Chaque version prend en charge de nouvelles fonctionnalités, mais si les utilisateurs ne mettent pas à jour leurs navigateurs cela devient un problème pour les webmasters comme vous qui créez des sites web. Chrome a résolu en grande partie le problème en mettant en place des mises à jour automatiques, sans intervention de l'utilisateur. Firefox a des utilisateurs qui ne pensent pas à se mettre à jour, et Internet Explorer a du mal à inciter à se mettre à jour car les dernières versions nécessitent aussi de mettre à jour Windows (Internet Explorer 9 n'est pas disponible pour Windows XP par exemple).

Des sites comme ***normansblog.de*** et ***caniuse.com*** tiennent notamment à jour une liste des fonctionnalités CSS supportées par les différentes versions de chaque navigateur :

### Les navigateurs sur mobile

Il faut savoir qu'il existe des variantes de ces navigateurs que l'on retrouve sur les téléphones portables, en particulier les *smartphones*.

De plus en plus de personnes consultent aujourd'hui des sites web sur leur portable, il faut donc connaître un minimum le fonctionnement des navigateurs des téléphones.



*Safari Mobile sur iPhone*

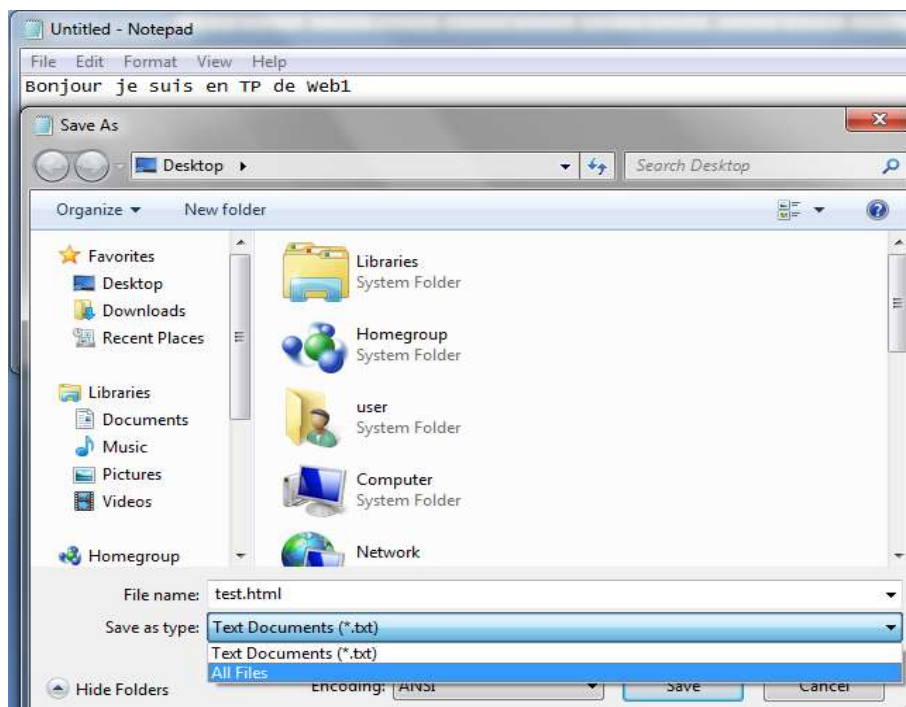
En fait, la plupart des navigateurs sur smartphones sont les mêmes que sur ordinateur, dans une version plus légère adaptée aux mobiles. Tout dépend du type de téléphone.

- **iPhone** : sur l'iPhone d'Apple, le navigateur utilisé est Safari Mobile. Il s'agit d'une version *light* mais néanmoins très complète de Safari pour ordinateur.
- **Android** : les portables sous Android bénéficient du navigateur Chrome Mobile. Là encore, il s'agit d'une version adaptée aux mobiles.
- **Windows Phone** : On retrouve Internet Explorer Mobile. Le principe est le même que pour les précédents navigateurs où il s'agit d'une version dédiée aux mobiles.
- **Blackberry** : les Blackberry font exception, car ils ont leur propre navigateur (il n'existe pas d'équivalent sur ordinateur). Néanmoins, les versions les plus récentes de ce navigateur se basent sur un noyau commun à Safari et Chrome (il s'agit du moteur de rendu Webkit). Par conséquent, l'affichage est en général proche de Safari et Chrome.

**L**es navigateurs pour mobiles supportent la plupart des dernières fonctionnalités de HTML et CSS. De plus, le système de mise à jour automatisé des mobiles nous garantit que les utilisateurs auront le plus souvent les dernières versions. Sachez néanmoins que des différences existent entre ces différents navigateurs mobiles et qu'il est conseillé aussi de tester son site sur ces appareils ; en particulier, l'écran étant beaucoup moins large, il faudra vérifier que votre site s'affiche correctement. Les tablettes tactiles sont équipées des mêmes navigateurs, l'écran étant simplement plus large.

## Chapitre II : Créer une page web avec l'éditeur

Dans la suite de ce cours, je travaillerai sous Notepad++. On va faire un petit essai. Je vous invite à écrire ce qui vous passe par la tête. Maintenant, enregistrons ce fichier **menu Fichier → Enregistrer**. Une boîte de dialogue vous demande où enregistrer le fichier et sous quel nom. Enregistrez-le où vous voulez. Donnez au fichier le nom que vous voulez, en terminant par .html, par exemple : *test.html*. Attention n'oubliez pas de changer le type en : *Tous les fichiers (All files)*



Ouvrez maintenant l'explorateur de fichiers dans le dossier où vous avez enregistré votre page. Vous y verrez le fichier que vous venez de créer. L'apparence du fichier dépend de votre navigateur web par défaut. Ici, l'icône est celle de Google Chrome, mon navigateur par défaut, mais le fichier a peut-être une autre icône chez vous.





Double-cliquez simplement sur ce fichier, votre navigateur s'ouvre en affichant le texte que vous avez écrit. Tout le texte s'affiche sur la même ligne et sans formatage. En fait, pour créer une page web il ne suffit pas simplement de taper du texte comme on vient de le faire. En plus de ce texte, il faut aussi écrire ce qu'on appelle des **balises**, qui vont donner des instructions à l'ordinateur comme "aller à la ligne", "afficher une image", etc.

## Les balises et leurs attributs

Il ne suffit pas "simplement" d'écrire du texte dans l'éditeur, il faut aussi donner des instructions à l'ordinateur. En HTML, on passe pour cela par des **balises**.

### Les balises

Les pages HTML sont remplies de ce qu'on appelle des balises. Celles-ci sont invisibles à l'écran pour vos visiteurs, mais elles permettent à l'ordinateur de comprendre ce qu'il doit afficher et comment le faire. Les balises se repèrent facilement. Elles sont entourées de "chevrons", c'est-à-dire des symboles < et >, comme ceci : **<balise>**. Elles indiquent la nature du texte autour d'elles. Elles veulent dire par exemple : *"Ceci est le titre de la page", "Ceci est une image", "Ceci est un paragraphe de texte", etc.*

On distingue deux types de balises : les balises en paires et les balises orphelines.

#### Les balises en paires

Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent:

**Code : HTML**

```
1 <title>Ceci est un titre</title>
```

On distingue une balise ouvrante (<title>) et une balise fermante (</title>) qui indique que le

titre se termine. Cela signifie pour l'ordinateur que tout ce qui n'est pas entre ces deux balises n'est pas un titre.

### Code : HTML

```
1 Ceci n'est pas un titre <title>Ceci est un titre</title> Ceci n'est pas un titre
```

### Les balises orphelines

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Une balise orpheline s'écrit comme ceci :

### Code : HTML

```
1 <head/>
```

Notez que le / de fin n'est pas obligatoire.

### Les attributs

Les attributs sont les options des balises. Ils viennent les compléter pour donner des informations supplémentaires. L'attribut se place après le nom de la balise ouvrante et a le plus souvent une valeur, comme ceci :

### Code : HTML

```
1 <balise attribut="valeur">
```

Exemple :

### Code : HTML

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Titre</title>
6   </head>
7
8   <body>
9
10  </body>
11 </html>
```

J'ai mis des espaces au début de certaines lignes pour "décaler" les balises. Ce n'est pas obligatoire et ça n'a aucun impact sur l'affichage de la page, mais ça permet de rendre le code source plus lisible. On appelle ça **l'indentation**.

Vous noterez que les balises s'ouvrent et se ferment dans un ordre précis. Par exemple, la balise `<html>` est la première que l'on ouvre, et c'est aussi la dernière que l'on ferme (tout à la fin du code, avec `</html>`). Les balises doivent être fermées dans le sens inverse de leur ouverture.

### Exemple :

- `<html><body></body></html>` : **correct**. Une balise qui est ouverte à l'intérieur d'une autre doit aussi être fermée à l'intérieur.
- `<html><body></html></body>` : **incorrect**, les balises s'entremêlent.

## Le doctype

### Code : HTML

```
1 <!DOCTYPE html>
```

La toute première ligne s'appelle le *doctype*. Elle indique qu'il s'agit bien d'une page web HTML. Ce n'est pas vraiment une balise comme les autres (elle commence par un point d'exclamation), vous pouvez considérer que c'est un peu l'exception qui confirme la règle. Cette ligne du doctype était autrefois incroyablement complexe. Il était impossible de la retenir de tête. Pour XHTML 1.0, il fallait écrire :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Pour HTML5, il a été décidé de la simplifier, pour le plus grand bonheur des webmasters. Quand vous voyez une balise doctype courte (`<!DOCTYPE html>`), cela signifie que la page est écrite en HTML5.

## La balise `<html>`

### Code : HTML

```
1 <html>
2
3
4 </html>
```

C'est la balise principale du code. Elle englobe tout le contenu de votre page. Comme vous pouvez le voir, la balise fermante `</html>` se trouve tout à la fin du code !

## L'en-tête `<head>` et le corps `<body>`

Une page web est constituée de 2 parties :

- L'en-tête `<head>` : cette section donne quelques informations générales sur la page, comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Cette section est généralement assez courte. Les informations que l'en-tête contient ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !
- Le corps `<body>` : c'est là que se trouve la partie principale de la page. Tout ce que nous écrirons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrirons la majeure partie de notre code.

## L'encodage (`charset`)

Code : HTML

```
1 <meta charset="utf-8" />
```

Cette balise indique l'encodage utilisé dans votre fichier *.html*. C'est lui qui détermine comment les caractères spéciaux vont s'afficher (accents, idéogrammes chinois et japonais, symboles arabes, etc.). Il y a plusieurs techniques d'encodage : ISO-8859-1, OEM 775, Windows-1253... Un seul cependant devrait être utilisé aujourd'hui autant que possible : UTF-8. Cette méthode d'encodage permet d'afficher sans aucun problème pratiquement tous les symboles de toutes les langues. Il faut aussi que votre fichier soit bien enregistré en UTF-8. C'est le cas le plus souvent sous Linux par défaut, mais sous Windows il faut généralement le dire au logiciel.

*Sous Notepad++, allez dans le menu format > Encoder en UTF-8*

Si vous avez un problème d'affichage des accents plus tard sur votre page web, c'est qu'il y a un problème avec l'encodage. Vérifiez que la balise indique bien UTF-8 et que votre fichier est enregistré en UTF-8 (votre éditeur de texte est capable de vous le dire, Notepad++ le fait dans le *menu format*).

## Le titre principal de la page

### Code : HTML

```
1 <title>
```

C'est le titre de votre page et qui décrit ce qu'elle contient. Il est conseillé que le titre soit assez court (moins de 100 caractères en général). Le titre ne s'affiche pas dans votre page mais dans l'onglet du navigateur.

## Insérer un commentaire

Un commentaire est une balise HTML avec une forme bien spéciale :

### Code : HTML

```
1 <!-- Ceci est un commentaire -->
```

Vous pouvez le mettre où vous voulez au sein de votre code source : il n'a aucun impact sur votre page, mais vous pouvez vous en servir pour vous aider à vous repérer dans votre code source.

### Code : HTML

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!-- En-tête de la page -->
5     <meta charset="utf-8" />
6     <title>Titre</title>
7   </head>
8
9   <body>
10    <!-- Corps de la page -->
11  </body>
12 </html>
```

**Tout le monde peut voir le code HTML de votre page** une fois celle-ci mise en ligne sur le Web. Il suffit de faire un clic droit sur la page et de sélectionner "Afficher le code source de la page". Cela s'explique assez facilement : le navigateur *doit* obtenir le code HTML pour savoir ce qu'il faut afficher. Le code HTML de tous les sites est donc public.

**P**ar conséquent, ne mettez pas d'informations sensibles comme des mots de passe dans les commentaires... et soignez votre code source

Tous les sites ne sont pas écrits en HTML5 (loin de là), et parfois certains webmasters écrivent très mal leur code, ce ne sont pas toujours des exemples à suivre

## Petit memo des balises les plus utilisées

Ces balises sont toutes situées dans l'en-tête de la page web, c'est-à-dire entre `<head>` et `</head>` :

Balise	Description
<code>&lt;link /&gt;</code>	<p>Cette balise permet d'indiquer certaines informations sur la page web. On l'utilise le plus souvent pour inclure une feuille de style CSS, comme ceci :</p> <p><b>Code : HTML</b></p> <pre>1 &lt;link rel="stylesheet" href="style.css" /&gt;</pre> <p>On peut aussi s'en servir pour 2-3 autres choses :</p> <p><b>Code : HTML</b></p> <pre>1 &lt;!-- Page d'accueil du site --&gt; 2 &lt;link rel="start" title="Accueil" href="index.html" /&gt; 3 &lt;!-- Page d'aide du site --&gt; 4 &lt;link rel="help" title="Politique d'accessibilité" href="accessibilite.html" /&gt; 5 &lt;!-- Fil RSS du site --&gt; 6 &lt;link rel="alternate" type="application/rss+xml" title="News de mon site" href="news.xml"/&gt; 7 &lt;!-- Icône du site (favicon) --&gt; 8 &lt;link rel="shortcut icon" type="image/x-icon" href="favicon.ico" /&gt;</pre> <p>La favicon est une icône qui s'affiche généralement à gauche de l'adresse de votre site sur le navigateur de vos visiteurs. C'est un moyen de personnaliser un peu plus son site. Quant au fil RSS, il s'agit d'une technique permettant à vos visiteurs de suivre l'actualité de votre site depuis un logiciel spécial (un navigateur tel que Firefox le fait d'ailleurs). En général on génère des fils RSS en PHP (si vous ne faites que du HTML / CSS ça ne vous intéresse donc pas pour le moment).</p>
<code>&lt;meta /&gt;</code>	<p>Cette balise permet de définir les propriétés de la page web. On s'en sert pour une foule de choses. Voici quelques exemples pratiques :</p> <p><b>Code : HTML</b></p> <pre>1 &lt;!-- Table de caractères --&gt; 2 &lt;meta charset="utf-8" /&gt; 3 &lt;!-- Auteur de la page --&gt; 4 &lt;meta name="author" content="Jean Dupont" /&gt; 5 &lt;!-- Description de la page --&gt; 6 &lt;meta name="description" content="La page personnelle de Jean Dupont" /&gt; 7 &lt;!-- Mots-clés de la page --&gt;</pre>

Balise	Description
	<pre> 8  &lt;meta name="keywords" content="expériences, recherche, laboratoire, chimie" /&gt; 9  &lt;!-- Adresse de contact --&gt; 10 &lt;meta name="reply-to" content="monadresse@email.com" /&gt; 11 &lt;!-- Empêcher la mise en cache de la page par le navigateur --&gt; 12 &lt;meta http-equiv="pragma" content="no-cache" /&gt; 13 &lt;!-- Rafraîchissement automatique au bout de 10 secondes --&gt; 14 &lt;meta http-equiv="refresh" content="10; URL=http://www.monsite.com" /&gt; </pre> <p>En général, on utilise surtout le meta pour la table de caractères.</p>
<script> >	<p>Permet de placer un script. On l'utilise souvent pour mettre du code Javascript :</p> <p><b>Code : HTML</b></p> <pre> 1  &lt;script&gt; 2  /* Votre script ici */ 3  &lt;/script&gt; </pre>
<style> >	<p>Permet de définir du code CSS pour la page.</p> <p>Exemple :</p> <p><b>Code : HTML</b></p> <pre> 1  &lt;style&gt; 2  /* Votre code CSS ici */ 3  &lt;/style&gt; </pre>
<title>	<p>Titre de la page web.</p> <p>C'est probablement la balise la plus importante d'une page web. Choisissez bien votre titre car il a beaucoup d'importance pour les moteurs de recherche (ils donnent de l'importance aux mots qui se trouvent dans le titre).</p> <p><b>Code : HTML</b></p> <pre> 1  &lt;title&gt;Les petites expériences informatiques de M. FAYE&lt;/title&gt; </pre>

## Balises de structuration du texte

Balise	Type	Description
<abbr>	<i>Inline</i>	Abréviation. Utiliser l'attribut title pour indiquer sa signification.
<blockquote>	<i>Block</i>	<p>Citation (longue)            Vous devez obligatoirement mettre une balise de paragraphe à l'intérieur du blockquote. Par exemple :</p> <p><b>Code : HTML</b></p> <pre> 1  &lt;blockquote&gt; 2      &lt;p&gt; 3          Texte de la citation 4      &lt;/p&gt; 5  &lt;/blockquote&gt;</pre>
<cite>	<i>Inline</i>	Citation du titre d'une oeuvre ou d'un évènement.
<q>	<i>Inline</i>	Citation (courte)
<sup>	<i>Inline</i>	Mise en exposant
<sub>	<i>Inline</i>	Mise en indice
<strong>	<i>Inline</i>	<p>Mise en valeur (forte)            Le texte est généralement mis en gras.</p>
<mark>	<i>Inline</i>	<p>Mise en valeur visuelle.            Le texte est généralement surligné.</p>
<em>	<i>Inline</i>	<p>Mise en valeur (faible)            Le texte est généralement mis en italique.</p>
<h6>	<i>Block</i>	Titre de niveau 6
<h5>	<i>Block</i>	Titre de niveau 5
<h4>	<i>Block</i>	Titre de niveau 4
<h3>	<i>Block</i>	Titre de niveau 3



Balise	Type	Description
<h2>	<i>Block</i>	Titre de niveau 2
<h1>	<i>Block</i>	Titre de niveau 1
<img />	<i>Inline</i>	<p>Insère une image. Utilisez les attributs <i>src</i> (pour indiquer l'adresse de l'image) et <i>alt</i> (pour indiquer un texte de remplacement). Ces 2 attributs sont obligatoires. Exemple :</p> <p><b>Code : HTML</b></p> <pre>1 &lt;img src="Templates/images/smiley.png" alt=":" /&gt;</pre>
<figure>	<i>Block</i>	Indique la présence d'une figure (image, code...) illustrant le texte.
<figcaption>	<i>Inline</i>	Description de la figure.
<audio>	<i>Inline</i>	Insère un son.
<video□>	<i>Inline</i>	Insère une vidéo.
<source>	-	Indique un format possible pour les balises <audio> et <video□>.
<a>	<i>Inline</i>	<p>Lien hypertexte. Indiquez l'url de destination grâce à l'attribut <i>href</i> :</p> <p><b>Code : HTML</b></p> <pre>1 &lt;a href="autrepage.html"&gt;Rendez-vous sur l'autre page&lt;/a&gt;</pre>
 	<i>Inline</i>	Retour à la ligne
<p>	<i>Block</i>	Paragraphe
<hr />	<i>Block</i>	Crée une ligne de séparation horizontale
<address>	<i>Block</i>	Permet d'indiquer une adresse, ou éventuellement l'auteur d'un document. Le texte est généralement mis en italique.

Balise	Type	Description
<del>	<i>Inline</i>	Permet d'indiquer un texte qui a été supprimé. Le texte est généralement barré.
<ins>	<i>Inline</i>	Permet d'indiquer un texte qui a été inséré. Le texte est généralement souligné.
<dfn>	<i>Inline</i>	Permet d'indiquer une définition.
<kbd>	<i>Inline</i>	Permet d'indiquer un code que doit taper le visiteur.
<pre>	<i>Block</i>	Le texte à l'intérieur de la balise <pre> sera affiché tel qu'il a été tapé dans le code (espaces et entrées compris). Une police de taille fixe est utilisée.
<progress>	<i>Inline</i>	Affiche une barre de progression. A utiliser conjointement avec les attributs value et max.
<time>	<i>Inline</i>	Pour indiquer une date ou une heure.

## Balises de liste

Cette partie énumère toutes les balises HTML permettant de créer des listes (listes à puces, listes numérotées, listes de définitions...)

Balise	Type	Description
<ul>	<i>Block</i>	<p>Liste à puces non numérotée. Vous devez mettre un &lt;li&gt;&lt;/li&gt; par élément de la liste. Exemple :</p> <p><b>Code : HTML</b></p> <pre>1 &lt;ul&gt; 2 &lt;li&gt;Un élément&lt;/li&gt; 3 &lt;li&gt;Un autre élément&lt;/li&gt; 4 &lt;/ul&gt;</pre>
<ol>	<i>Block</i>	<p>Liste à puces numérotée. Vous devez mettre un &lt;li&gt;&lt;/li&gt; par élément de la liste. Exemple :</p>

Balise	Type	Description
		<b>Code : HTML</b> <pre> 1  &lt;ol&gt; 2  &lt;li&gt;Elément n°1&lt;/li&gt; 3  &lt;li&gt;Elément n°2&lt;/li&gt; 4  &lt;/ol&gt;</pre>
<li>	<i>list-item</i>	Permet de créer un élément de liste. Le type de la balise est particulier car elle n'est ni block ni inline. On dit qu'elle est de type <i>list-item</i> .
<dl>	<i>Block</i>	Liste de définitions. Vous devez alterner chaque terme <dt> par sa définition <dd>. Exemple :  <b>Code : HTML</b> <pre> 1  &lt;dl&gt; 2  &lt;dt&gt;Porte&lt;/dt&gt; 3  &lt;dd&gt;Ouverture dans un mur permettant d'entrer et de sortir&lt;/dd&gt; 4  &lt;dt&gt;Théâtre&lt;/dt&gt; 5  &lt;dd&gt;Lieu où l'on représente des ouvrages dramatiques&lt;/dd&gt; 6  &lt;/dl&gt;</pre>
<dt>	<i>Block</i>	Terme à définir
<dd>	<i>Block</i>	Définition du terme

## Balises de tableau

Balise	Type	Description
<table>	<i>Block</i>	Délimite un tableau. Voici un exemple de tableau simple :  <b>Code : HTML</b> <pre> 1  &lt;table&gt; 2  &lt;caption&gt;Passagers du vol 377&lt;/caption&gt;</pre>

Balise	Type	Description
		<pre> 3 4  &lt;tr&gt; 5    &lt;th&gt;Nom&lt;/th&gt; 6    &lt;th&gt;Age&lt;/th&gt; 7    &lt;th&gt;Pays&lt;/th&gt; 8  &lt;/tr&gt; 9 10 &lt;tr&gt; 11   &lt;td&gt;Carmen&lt;/td&gt; 12   &lt;td&gt;33 ans&lt;/td&gt; 13   &lt;td&gt;Espagne&lt;/td&gt; 14 &lt;/tr&gt; 15 &lt;tr&gt; 16   &lt;td&gt;Michelle&lt;/td&gt; 17   &lt;td&gt;26 ans&lt;/td&gt; 18   &lt;td&gt;Etats-Unis&lt;/td&gt; 19 &lt;/tr&gt; 20 &lt;tr&gt; 21   &lt;td&gt;François&lt;/td&gt; 22   &lt;td&gt;43 ans&lt;/td&gt; 23   &lt;td&gt;France&lt;/td&gt; 24 &lt;/tr&gt; 25 &lt;/table&gt; </pre>
<caption>	-	Permet de donner un titre au tableau
<tr>	-	Ligne de tableau
<th>	-	Cellule d'en-tête du tableau (généralement mise en gras)
<td>	-	Cellule du tableau
<thead>	-	<p>Balise non obligatoire permettant d'insérer l'en-tête du tableau. Si vous choisissez d'utiliser &lt;thead&gt;, &lt;tfoot&gt; et &lt;tbody&gt;, vous devez les mettre dans l'ordre suivant dans votre code source :</p> <ol style="list-style-type: none"> <li>1. &lt;thead&gt;</li> <li>2. &lt;tfoot&gt;</li> <li>3. &lt;tbody&gt;</li> </ol>

Balise	Type	Description
<tbody>	-	Balise non obligatoire permettant d'insérer le corps du tableau
<tfoot>	-	Balise non obligatoire permettant d'insérer le pied du tableau

## Balises de formulaire

Balise	Type	Description
<form>	<i>Block</i>	<p>Délimite un formulaire. Vous devrez généralement donner 2 attributs à la balise &lt;form&gt;</p> <ul style="list-style-type: none"> <li><b>method</b> : indique la méthode d'envoi du formulaire (get ou post). Si vous ne savez pas quoi utiliser, mettez post.</li> <li><b>action</b> : la page vers laquelle le visiteur doit être redirigé lorsqu'il a validé votre formulaire.</li> </ul>
<fieldset>	<i>Block</i>	<p>Permet de regrouper plusieurs éléments d'un formulaire. On l'utilise généralement dans de grands formulaires.</p> <p>Pour donner un titre à votre groupe, utilisez la balise &lt;legend&gt;</p>
<legend>	<i>Inline</i>	<p>Titre d'un groupe dans un formulaire. A utiliser à l'intérieur d'un &lt;fieldset&gt;</p>
<label>	<i>Inline</i>	<p>Titre d'un élément de formulaire. Généralement, vous devrez mettre l'attribut <i>for</i> sur cette balise pour indiquer l'ID de l'élément auquel correspond le label.</p>
<input />	<i>Inline</i>	<p>Champ de formulaire. Il existe de nombreux types de champs différents. Vous choisissez le type de champ que vous désirez grâce à l'attribut <i>type</i> :</p> <p><b>Code : HTML</b></p> <pre> 1  &lt;!-- Zone de texte d'une ligne --&gt; 2  &lt;input type="text" /&gt;</pre>

Balise	Type	Description
		<pre> 3  &lt;!-- Mot de passe (le texte est caché) --&gt; 4  &lt;input type="password" /&gt; 5  &lt;!-- Envoi de fichier --&gt; 6  &lt;input type="file" /&gt; 7  &lt;!-- Case à cocher --&gt; 8  &lt;input type="checkbox" /&gt; 9  &lt;!-- Bouton d'option --&gt; 10 &lt;input type="radio" /&gt; 11 &lt;!-- Bouton --&gt; 12 &lt;input type="button" /&gt; 13 &lt;!-- Bouton d'envoi --&gt; 14 &lt;input type="submit" /&gt; 15 &lt;!-- Bouton de remise à zéro --&gt; 16 &lt;input type="reset" /&gt; 17 &lt;!-- Champ caché --&gt; 18 &lt;input type="hidden" /&gt; </pre> <p>Pensez à donner un nom à vos champs grâce à l'attribut <i>name</i></p>
<textarea>	Inline	<p>Zone de saisie multiligne.</p> <p>Vous pouvez définir sa taille grâce aux attributs <i>rows</i> et <i>cols</i> (nombre de lignes et colonnes) ou bien le faire en CSS grâce aux propriétés <i>width</i> et <i>height</i>.</p>
<select>	Inline	<p>Liste déroulante.</p> <p>Utilisez la balise &lt;option&gt; pour créer chaque élément de la liste :</p> <p><b>Code : HTML</b></p> <pre> 1  &lt;select name="pays"&gt; 2    &lt;option value="france"&gt;France&lt;/option&gt; 3    &lt;option value="espagne"&gt;Espagne&lt;/option&gt; 4    &lt;option value="italie"&gt;Italie&lt;/option&gt; 5  &lt;/select&gt; </pre>
<option>	Block	Element d'une liste déroulante
<optgroup>	Block	<p>Groupe d'éléments d'une liste déroulante.</p> <p>A utiliser dans le cas d'une grande liste déroulante.</p> <p>Vous devez utiliser l'attribut <i>label</i> pour donner un nom au groupe.</p>

## Balises sectionnantes

Ces balises permettent de construire le squelette de notre site web :

Balise	Type	Description
<header>	<i>Block</i>	En-tête
<nav>	<i>Block</i>	Liens principaux de navigation
<footer>	<i>Block</i>	Pied de page
<section>	<i>Block</i>	Section de page
<article>	<i>Block</i>	Contenu ayant un sens propre (billet de blog, actualité...) pouvant être repris sur un autre site.
<aside>	<i>Block</i>	Informations complémentaires

## Balises génériques

Les balises génériques sont des balises qui n'ont pas de sens sémantique. En effet, toutes les autres balises HTML ont un **sens** : <p> signifie "Paragraphe", <h2> signifie "Sous-titre" etc. Parfois, on a besoin d'utiliser des balises génériques (aussi appelées *balises universelles*) car aucune des autres balises ne convient. On utilise le plus souvent des balises génériques pour construire son design. Il y a 2 balises génériques : l'une est inline, l'autre est block.

Balise	Type	Description
<span>	<i>Inline</i>	Balise générique de type inline
<div>	<i>Block</i>	Balise générique de type block

Ces balises ont un intérêt uniquement si vous leur donnez un attribut *class*, *id* ou *style* :

- **class** : indique le nom de la classe CSS à utiliser.
- **id** : donne un nom à la balise. Ce nom doit être unique sur toute la page car il permet d'identifier la balise. Vous pouvez vous servir de l'ID pour de nombreuses choses, comme par exemple pour un lien vers une ancre, pour un style CSS de type ID, pour des manipulations en Javascript etc.
- **style** : cet attribut vous permet d'indiquer directement le code CSS à appliquer. Vous n'êtes donc pas obligés d'avoir une feuille de style à part, vous pouvez juste mettre directement les attributs CSS. Notez qu'il est préférable de ne pas utiliser cet attribut et de passer à la place par une feuille de style externe car cela rend votre site plus facile à mettre à jour par la suite.

Ces 3 attributs ne sont pas réservés aux balises génériques : vous pouvez aussi les mettre sur la plupart des autres balises sans aucun problème.

En théorie, on pourrait faire presque tout un site rien qu'avec les balises génériques <div> et <span> (en utilisant du CSS), mais votre site n'aurait aucun sens logique ! Or, respecter la logique de son code source est une chose que les webmasters considèrent comme fondamentale. Une page sémantique a notamment plus de chances d'être mieux indexée dans Google qu'une page utilisant des balises inadaptées.