

## Table des matières

Plantons le décor ! .....	1
Concepts de base du HTML .....	2
1. Les liens hypertexte .....	2
2. Les balises.....	2
3. Ossature d'une page HTML .....	3
Une première page web .....	5
1. Les balises indispensables .....	6
2. Balises bloc, balises en ligne .....	9
3. Hiérarchie et propriétés des blocs d'informations .....	10
Mise en page et feuilles de style .....	11
1. Structure d'une feuille de style .....	11
L'écriture des déclarations :.....	13
2. Unités de mesures et couleurs.....	14
2.1 Mesures absolues .....	14
2.2 Mesures relatives .....	14
2.3 Indications de couleurs .....	15
3. Positionnement des blocs.....	15
3.1. Marges externes, marges internes et dimensions.....	15
3.2. Positionnement flottant.....	18
3.3. Positions absolues, fixes et relatives.....	19
4. Cascade de styles et priorités des styles .....	20
Ecriture des pages d'un site .....	21
1. Matériel nécessaire .....	21
2. Etapes de la fabrication d'un site.....	21
Créer des menus simples .....	22
1. Un menu vertical.....	22
2. Un menu horizontal .....	24
Maîtriser les liens.....	27
Liens internes à une page.....	27
Liens vers une page différente .....	27
Liens externes au site.....	28
Liens e-mail .....	28
Balises HTML .....	29
1. Les balises bloc.....	29
2. Les balises en ligne.....	30
Propriétés CSS .....	31
1. Marges .....	31
2. Positionnement des blocs.....	31
3. Alignement de texte.....	33
4. Formatage d'écriture.....	33
5. Bordures.....	34
6. Arrières-plans .....	34
7. Formatage des listes.....	35
8. Formatage des tableaux .....	36
9. Les media queries (requêtes de media).....	36
Glossaire Internet.....	37
Codification des caractères spéciaux.....	44

## Plantons le décor !

Visiter un site web, c'est un peu comme aller au théâtre. On peut d'ailleurs tenter d'établir un parallèle entre ces deux spectacles :

- La pièce de théâtre / le site consulté
- Les différents actes / les pages du site
- Le livret / le code source des pages
- Le producteur / Internet
- La scène / l'écran de l'ordinateur
- Le metteur en scène / le navigateur
- Le public / l'ensemble des internautes.

Les pièces de théâtre du producteur Internet sont interprétées sur toutes les scènes de notre planète. L'écriture du livret utilise donc un langage reconnu par tous les metteurs en scène :

le **HTML** (Hyper Text Markup Language).

Le *World Wide Web Consortium*, plus connu sous l'acronyme *W3C* est l'organisme qui définit les normes du HTML et ses évolutions. La dernière version est appelée **HTML5**.

S'il est normal - comme au théâtre - que le résultat final diffère selon le metteur en scène, l'auteur (celui qui a conçu le site) doit veiller à ce que le metteur en scène respecte le livret et ne dénature pas l'œuvre. Nous verrons en effet que tous les navigateurs ne traduisent pas le code source HTML dans un parfait respect des normes qui le définissent...

Un fichier contenant du code source HTML se caractérise par l'extension **.html** .

## Concepts de base du HTML

Le langage d'écriture des pages web permet d'implanter des liens vers d'autres pages, de décrire la forme du texte (gras, italique, polices, ...), comme les premiers traitements de texte et d'inclure des images fixes ou animées, du son, de la vidéo et même des programmes interactifs (à l'aide de Javascript).

Le langage HTML est également utilisé pour fournir une interface aux CD ROM multimédia et aux DVD ROM.

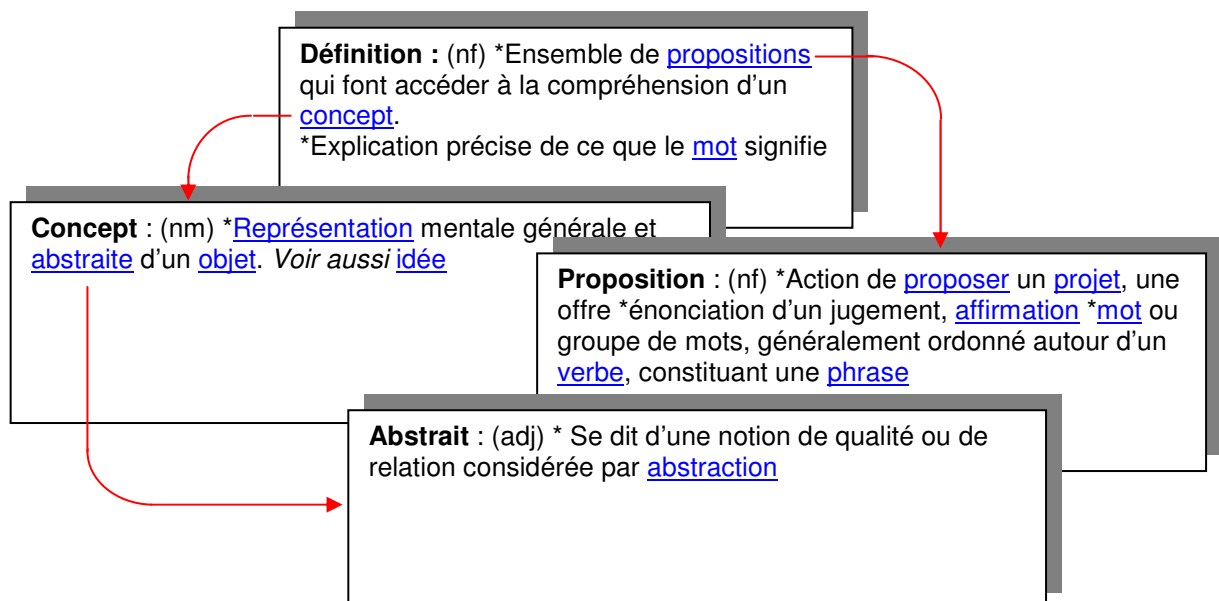
### 1. Les liens hypertexte

Ils constituent la base de la navigation sur les sites de la toile mondiale.

Un *lien hypertexte* peut être constitué d'un mot (ou d'un ensemble de mots) généralement souligné qui réagit avec la souris et renvoie automatiquement vers un endroit appelé *cible* qui traite le sujet.

La cible du lien peut se situer dans la page courante, dans une autre page du même site ou dans un site différent.

Un dictionnaire sur le web illustre bien la notion de lien hypertexte :



### 2. Les balises

Les balises constituent le principe même du HTML (langage à balises).

Les navigateurs permettent de visualiser le code source correspondant à une page affichée.

Dans le code source, une balise (ou tag) est facilement identifiable : elle est constituée d'un ou plusieurs mots encadrés par les signes inférieur (<) et supérieur (>).

Exemple : `<p align="center"> Fabriquer un site web </p>`

Balise d'ouverture ↑
texte
Balise de fermeture ↑

*p* est un élément du langage HTML.  
*align* est un attribut de l'élément *p*.  
 ="center" affecte la valeur *center* à l'attribut *align*.  
 Certains éléments peuvent avoir plusieurs attributs.

**Chaque balise doit être ouverte puis fermée** ; les balises `<br />` `<hr />` ou `<img ... />` (respectivement saut de ligne, trait horizontal et insertion d'une image) sont parmi les rares exceptions. On parle alors de balises "auto-fermantes".

Bien que majuscules et minuscules soient autorisées, **les minuscules** seront toujours utilisées dans ce document pour être conforme au XHTML (eXtended Hypertext Markup Language) et assurer la compatibilité avec les évolutions du langage.

### 3. Ossature d'une page HTML

```

<!DOCTYPE ..... >
  <!-- Déclaration du standard HTML utilisé -->

<html>
  <head>
    <!-- En-tête du document -->
  </head>

  <body>
    <!-- Ici figurent l'ensemble des informations à afficher -->
  </body>
</html>

```

Commentaires  
 <!--                    -->

#### Important:

- Les balises `<head>` et `<body>` sont imbriquées dans la balise `<html>`.

Dans les pages HTML l'imbrication des balises est **systématique**.

L'indentation (non obligatoire) permet de s'assurer du respect de la **règle absolue** :

La première balise rencontrée dans le source est aussi la dernière fermée.  
 Réciproquement la dernière balise rencontrée est la première fermée.

Un exemple d'écriture incorrecte :

```
<html>
<body>
.....
</html>
</body>
```

- Les commentaires ne sont pas obligatoires et leur couleur est sans importance.  
Cependant la présence d'un commentaire peut faciliter la compréhension du code source et les modifications de la page...

### ➤ Déclaration du standard HTML utilisé

La déclaration du "**doctype**" qui spécifie le standard HTML utilisé dans la rédaction des pages de site n'est pas obligatoire. Elle est nécessaire pour effectuer auprès du W3C une validation qui consiste à vérifier le respect des normes d'écriture du standard choisi. Elle aide aussi le navigateur à interpréter la page le plus correctement possible.

**Dans ces feuillets, il sera toujours question de XHTML**, qui est la dernière version du langage HTML. D'où la "formule" doctype correspondante :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
```

### ➤ L'en-tête du document (<head>)

Après la balise **<html>**, début de tout code source, la balise **<head>** donne des informations générales :

- la balise **<title>** donne le titre de votre page tel qu'il apparaît au-dessus de la fenêtre du navigateur ou dans les favoris du visiteur qui rajoute votre site à sa liste de favoris.
- les balises **<meta />** :

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF8" />
```

précise le jeu de caractères utilisés pour l'affichage des textes.

```
<meta name="keywords" content="festival, gospel, midi-pyrénées, france" />
```

énumère des mots clés judicieux permettant aux moteurs de recherche (Google et autres) de référencer votre site.

- les balises **<link />** :

```
<link rel="stylesheet" href="habillage.css" type="text/css" media="screen" />
```

donne le lien du code avec une feuille de styles réalisant la mise en page. Il peut y avoir plusieurs fichiers CSS pour une page.

### ➤ Le corps du document (<body>)

C'est ici que se trouvent les données à afficher (titres, textes, images, tableaux, liens...)

## Une première page web

Pour écrire le code HTML un éditeur élémentaire tel que le *Bloc notes* sous Windows convient.

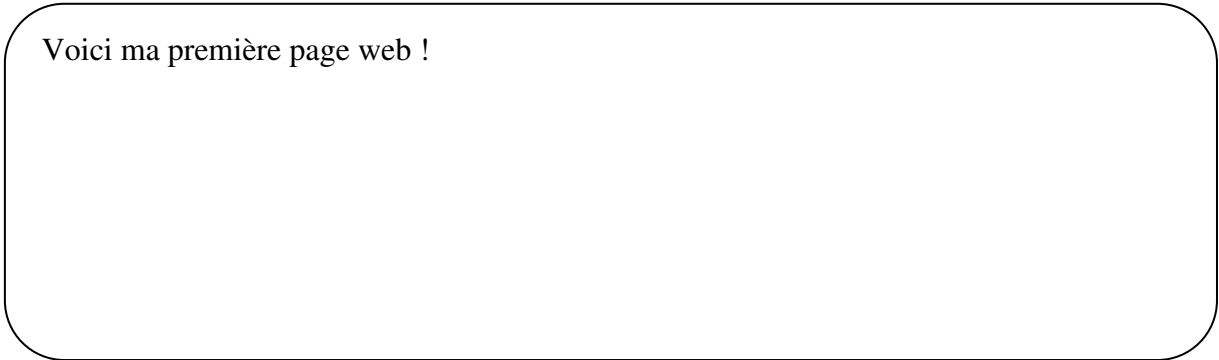
Exemple de page minimale (sans doctype ni balises meta) :

```
<html>
<head>  <title> Ma première page</title>  </head>
<body>
    Voici ma première page web!
</body>

</html>
```

Il faut ensuite enregistrer la page sur le disque dur en lui donnant un nom de fichier. Une page d'accueil de site se nomme généralement "**index.html**".

Dès lors, un double-clic sur le nom du fichier source de cette page suffira à ouvrir le navigateur Internet et à afficher à l'écran :



Voici ma première page web !

Ce premier succès encourage à composer une page avec un peu plus de contenu. Il est possible de faire un copier-coller pour récupérer une partie d'un document **.doc** ou **.rtf** et l'insérer à la place de la phrase actuelle.

Hélas l'affichage de la nouvelle page par le navigateur ne reflète pas exactement la mise en forme du texte:

- les passages à la ligne suivante et les tabulations sont ignorés.
- les espaces consécutifs sont réduits à un espace unique.

Ce n'est pas un dysfonctionnement, c'est la règle avec le langage HTML !

Un certain nombre de balises seront indispensables pour rendre au texte sa structure.

## 1. Les balises indispensables

### La balise p

Elle indique un paragraphe (au sens littéraire) et provoque un saut de ligne.

Exemple:

```
<p>Le premier paragraphe </p>
<p>Le deuxième paragraphe </p>
<p>Le troisième paragraphe</p>
```

Résultat :

```
Le premier paragraphe
Le deuxième paragraphe
Le troisième paragraphe
```

### La balise br

Elle permet un simple passage à la ligne suivante sans saut de ligne.

Exemple:

```
<p> /* Début du paragraphe */
  La première phrase <br />
  /* ligne suivante */
  La deuxième <br />
  /* ligne suivante */
  La troisième
</p> /* Fin du paragraphe */
Le paragraphe est fini !
```

Résultat :

```
La première phrase
La deuxième
La troisième

Le paragraphe est fini !
```

### La balise a

Indique un lien hypertexte

Exemple:

```
Visitez notre <a href="page.html"
title="Nos activités">Page web</a>
```

Résultat :

```
Visitez notre Page web
```

Attributs associés à la balise a :

**href :** indique l'adresse (relative ou absolue – cf répertoires et arborescence) de la cible vers laquelle le lien renvoie.  
La cible peut être une page du site (cas des menus) ou d'un autre site.

[Page web](#) est le lien hypertexte vers "page.html", fichier situé dans le même répertoire que le source.

**title :** Permet de spécifier une description du lien.

Au passage de la souris sur le lien, le navigateur affichera "Nos activités" dans une bulle.

Le lien décrit par l'attribut **href** peut être :

- interne à une page web (cas du sommaire d'une page)
- vers une page différente (cas d'un menu général de site)
- vers un site différent
- vers un e-mail (Exemple écrire au webmestre : ouverture automatique du logiciel de messagerie avec l'adresse de destination déjà remplie).

Les divers cas de figure sont détaillés page 25: [Maîtriser les liens](#).

## La balise img

Insertion d'une image

Exemple:

```
<p>
  Les singes et les hommes
  
</p>
```

Résultat :



Attributs associés :

**src :** Attribut **obligatoire** qui donne l'adresse de l'image (relative ou absolue – cf répertoires et arborescence).

**alt :** Texte alternatif qui permet aux déficients visuels utilisant un logiciel de lecture audio d'avoir une interprétation de l'image.



## La balise ul

Délimite une liste à puces

Exemple:

```
<p>Voici une liste :</p>
<ul>
  <li> Rubrique1 </li>
  <li> Rubrique2 </li>
  <li> Rubrique3 </li>
  <li> Rubrique4 </li>
</ul>
```

Résultat :

Voici une liste:

- Rubrique1
- Rubrique2
- Rubrique3
- Rubrique4

Les items de la liste sont placés à l'intérieur de balises `<li>` et s'affichent précédés d'une puce.

## Les balises h1 à h6

Titre hiérarchisés du niveau 1 (le plus gros) au niveau 6 (le plus petit)

Exemple:

```
<h1>Titre général</h1>
<h2>Titre de niveau 2</h2>
<h3>Titre de niveau 3</h3>
<h2>Un autre titre de niveau 2</h2>
```

Résultat :

**Titre général**  
**Titre de niveau 2**  
 Titre de niveau 3  
**Un autre titre de niveau 2**

## La balise strong

Caractères gras

Exemple:

```
<p>
  Il faut définir ce que sont
  <strong>les balises</strong>
</p>
```

Résultat :

Il faut définir ce que sont  
**les balises**

## La balise em

### Caractères obliques

#### Exemple:

```
<p>
  Nous devons revenir sur <em>les
  liens hypertexte</em> car ils sont
  la base de la navigation sur
  Internet.
</p>
```

#### Résultat :

Nous devons revenir sur *les liens hypertexte* car ils sont la base de la navigation sur Internet.

## La balise div

Division de texte. Cette balise générique ne produit par défaut aucun effet particulier. Elle sert à diviser le texte en bloc ayant une unité sémantique et (ou) une mise en page particulière.

#### Exemple:

```
<div>
  Dans ce chapitre nous aborderons
  les arguments en faveur des
  documents bien structurés, et de
  la mise en page soignée qui rend
  leur consultation plus agréable.
</div>
```

#### Résultat :

Dans ce chapitre nous aborderons les arguments en faveur des documents bien structurés, et de la mise en page soignée qui rend leur consultation plus agréable.

## 2. Balises bloc, balises en ligne

Les balises possèdent deux types de comportement différent ;

- **Les balises "bloc"** qui servent à distinguer les parties entières de texte, comme des titres, des paragraphes, des listes, des divisions (**h1** à **h6**, **p**, **ul**, **div**).  
A l'affichage, les "blocs" se placent toujours par défaut l'un en dessous de l'autre (comme le ferait un retour chariot).
- **Les balises "en ligne"** insérées dans le texte pour l'enrichir comme un lien hypertexte, italique, gras, image (**a**, **em**, **strong**, **img**).  
Elles interviennent dans le déroulement naturel du texte sans provoquer de saut de ligne.

Les balises bloc peuvent servir de **conteneur** à du texte, des balises en ligne ou - exception faite des balises de titre h1 à h6 - à d'autres balises bloc.

Les balises en ligne **ne peuvent contenir que des balises en ligne**.

Exemples :

- `<b> <div> blabla </div> </b>` est incorrect
- `<div> <b> blabla </b> </div>` est correct

### 3. Hiérarchie et propriétés des blocs d'informations

Un document HTML est toujours composé de conteneurs.

L'imbrication des différents conteneurs compose une structure hiérarchique semblable à une famille et conduit à une hérédité : la transmission des caractéristiques d'un conteneur aux blocs qu'il contient.

Voici la définition des termes qui seront employés :

- Un bloc contenu directement dans un conteneur est dit *Enfant* de ce conteneur.
- Le bloc contenant directement un bloc contenu est dit *Parent* de ce bloc.
- Un élément *Ancêtre* est un élément bloc qui contient un autre élément ou une hiérarchie d'éléments.  
Exemple : l'élément *body* est l'*Ancêtre* de tous les éléments d'une page.
- A défaut de nouvelles propriétés définies pour un de ces blocs, tous les éléments d'une page *héritent* des propriétés du bloc *body*.
- Un bloc *Parent* est l'*Ancêtre* immédiat du bloc qu'il contient directement.
- Les éléments ayant le même élément *Parent* sont appelés *Frères*.
- Tout bloc inséré dans un conteneur *hérite*, par défaut, des propriétés définies pour le conteneur.

Pour donner aux blocs leurs propriétés (positions à l'intérieur de la page, dimensions, couleur de fond, police de caractères, etc...) on utilisera *des feuilles de style* (semblables à celles des traitements de texte) qui seront associées au code HTML par l'intermédiaire de balises **link** (voir page 4 : [Ossature d'une page HTML](#)) situées dans l'entête du document.

Avant l'introduction des feuilles de style, les mises en pages étaient réalisées en utilisant des tableaux, souvent imbriqués et modifier une page était un exercice véritablement fastidieux.

## Mise en page et feuilles de style

Dans la construction d'un site web, l'utilisation des feuilles de styles en cascade, appelées communément CSS (*Cascading Style Sheets*), permet de séparer le fond de la forme.

Le source HTML contient **uniquement** le texte et les images structurés par les balises. Si le code est bien écrit, un site doit pouvoir être consulté sans avoir de feuilles de styles (une vérification préliminaire à effectuer...).

Les CSS associées au code source donnent **toutes** les informations de présentation utilisées dans les pages :

- police utilisées dans les titres, les blocs d'information, les listes
- couleurs de fond ou du texte
- bordures de texte
- positionnement des blocs, marges internes et externes
- etc.

Les feuilles de styles sont consignées dans un fichier caractérisé par l'extension **.css**

Le source HTML contient dans l'entête l'indication de liaison avec la CSS associé :

```
<head>
  <meta..... />
  <title>Fabriquer un site web</title>
  <link rel="stylesheet" href="habillage.css" type="text/css" />
</head>
```

Le fichier CSS s'appelle ici "**habillage.css**"

### 1. Structure d'une feuille de style

**sélecteur{propriété:valeur;}      Exemple : p { font-size : 14pt ; }**

*p* est le *sélecteur* de la feuille de style composé d'un seul *élément*. C'est un *sélecteur simple*.

*font-size* est une *propriété* de la feuille de style.

*14pt* est la *valeur* affectée à la propriété font-size.

Chaque couple **propriété:valeur** terminé par **;** est une *déclaration*.

Une feuille de style peut définir autant de *déclarations* que le sélecteur peut autoriser.

Une **propriété** correspond à un attribut d'élément HTML **de type bloc** (`margin-top` , `font-size`, `background-color`, etc.)

Une **valeur** doit être cohérente avec la propriété CSS à laquelle elle est affectée (par exemple une couleur pour `background-color` et pas une mesure). Dans le cas contraire, la déclaration est ignorée.

Des commentaires peuvent être insérés en tout endroit du fichier sous la forme :

```
/* Ceci est un commentaire */
```

Des *sélecteurs contextuels*, formés par combinaisons de sélecteurs simples, permettent de viser des éléments selon leur position dans le document :

Exemple : **a img** ne désignera que les images présentes dans des liens.

On distingue 3 catégories de feuilles de style :

- *Les feuilles de styles globales :*

Le nom d'élément appartient aux balises bloc du HTML (h1, p, div, td, etc...).

Exemple : **p { color : green ; }**

Résultat :

- Cette feuille de style va affecter la couleur verte à **tous les paragraphes de la page.**

- *Les feuilles de style indépendantes :*

Le nom d'élément, appelé identifiant, est indépendant des balises bloc du HTML, est de la forme **#nom** .

Ce nom, choisi librement, affectera des **caractéristiques individuelles à un bloc unique.**

Une dénomination significative est à privilégier.

Exemple : **#entete { font-size : 14pt ; }**

Côté HTML : `<div id="entete">...</div>`

Résultat :

- Le texte placé à l'intérieur de ce conteneur aura une taille de police de 14pt.
- Tout autre bloc **div** non contenu dans le bloc `id="entete"` **n'aura pas la taille de police précisée dans la feuille de style #entete.**

- *Les définitions de classes :*

Ces feuilles de style sont repérables par la forme de leur nom **. nom**.

A la différence de sa cousine la feuille de style indépendante, une classe pourra affecter des **caractéristiques identiques à plusieurs blocs.**

Exemple : **.encart { border : solid ; }**

Côté HTML : `<div class="encart">...</div>`

Résultat :

- Le texte placé à l'intérieur de ce conteneur sera entouré d'une bordure continue.
- Toutes les pages du site liées à notre fichier CSS, auront les blocs **div** appartenant à la classe `"encart"` entourés d'une bordure continue.

## Les pseudo-classes :

Elles sont principalement employées associées derrière l'élément a (lien). La syntaxe est de la forme :

**élément:pseudo-classe{propriété:valeur ; ...}**

*Exemples :*

```
a:link {color:blue ;}
```

Tous les liens vers des cibles non visitées seront écrits en bleu

```
a:visited {color:purple ;}
```

Tous les liens vers des cibles déjà visitées seront écrits en violet.

```
a:hover {text-decoration:underline;}
```

Au survol d'un lien par le pointeur de la souris le libellé sera souligné.

*La pseudo-classe :hover est ignorée d'IE6 lorsqu'elle n'est pas associée aux liens.*

```
a:active {color:red;}
```

Sélectionne le lien activé lorsqu'on le clique avec la souris.

```
a:focus { propriété CSS:valeur; ... }
```

Sélectionne un lien qui reçoit la zone de saisie active d'un formulaire.

*La pseudo-classe :focus est ignorée de Google Chrome et Apple Safari et n'est pas supportée par Internet Explorer avant la version 8.*

## Les pseudo-éléments :

*Exemples :*

```
p:first-letter {font-size:200%;}
```

La première lettre d'un paragraphe aura une taille de police multipliée par 2.

```
p:first-line {text-decoration:overline;}
```

La première ligne d'un paragraphe sera placée au-dessous d'une ligne horizontale.

## L'écriture des déclarations :

Elle doit répondre à certains impératifs :

- faciliter la lecture des feuilles de styles par les utilisateurs ce qui conduit à multiplier les propriétés individuelles.
- faciliter leur écriture en recourant à un nombre plus réduit de propriétés combinées.

Pour répondre à ces contraintes opposées, la définition des marges, des fontes, des bordures, des arrière-plans et du style des listes, peut se faire selon deux modes : la notation classique et la notation globale.

## Exemples : Définir les caractéristiques d'une fonte

<pre> /* Notation classique */  .sous_titre {   font-size : small;   font-family : Times;   font-weight : bold;   font-style : italic; } </pre>	<pre> /* Notation globale */  .sous_titre {   font : Times small bold italic; } </pre>
---	--

## 2. Unités de mesures et couleurs

Le signe décimal des unités de mesures est le point.

### 2.1 Mesures absolues

<b>pt</b>	Point. Mesure typographique. 1 point correspond à 1/72 de pouce
<b>pc</b>	Pica. Unité de mesure typographique. 1 pica correspond à 12 points
<b>px</b> <sup>1</sup>	Pixel. Mesure absolue pour un écran donné, mais relative à la définition de l'affichage si on modifie ce dernier
<b>in</b>	1 pouce correspond à 2,54 centimètres
<b>mm</b>	Millimètre. Seules les valeurs entières sont autorisées
<b>cm</b>	Centimètre

### 2.2 Mesures relatives

C'est ce type de mesures qu'il faut privilégier car elles harmonisent taille des caractères et taille disponible pour l'affichage.

<b>em</b>	Relatif à la taille de la lettre <b>m</b> dans la police de l'élément
<b>ex</b>	Relatif à la taille de la lettre <b>x</b> dans la police de l'élément
<b>%</b>	Pourcentage relatif à la taille disponible pour l'élément. Cette unité est intéressante pour adapter l'espace occupé par un bloc de texte à la définition de l'écran de l'ordinateur (1280 X 800, 1024 X 768, 800 X 600, etc...)

---

<sup>1</sup> **Attention** : De IE 5.0 à IE 7.0 impossible à l'utilisateur de modifier la taille de la police affichée dont l'unité est le pixel

## 2.3 Indications de couleurs

16,7 millions de couleurs sont disponibles ( $256^3$ ):

#xxxxxx	Mention hexadécimale RGB comme pour HTML (ex FFFFFFF)
rgb(R, G, B)	Valeurs RGB décimales (0-255,0-255,0-255)
rgb(%, %, %)	Répartition RGB en pourcentage (0-100%,0-100%,0-100%)

Couleur (16 noms standards)	 Aqua	 Red	 Yellow	 White
	 Blue	 Purple	 Lime	 Silver
	 Navy	 Maroon	 Olive	 Gray
	 Fuchsia	 Green	 Teal	 Black

## 3. Positionnement des blocs

### 3.1. Marges externes, marges internes et dimensions

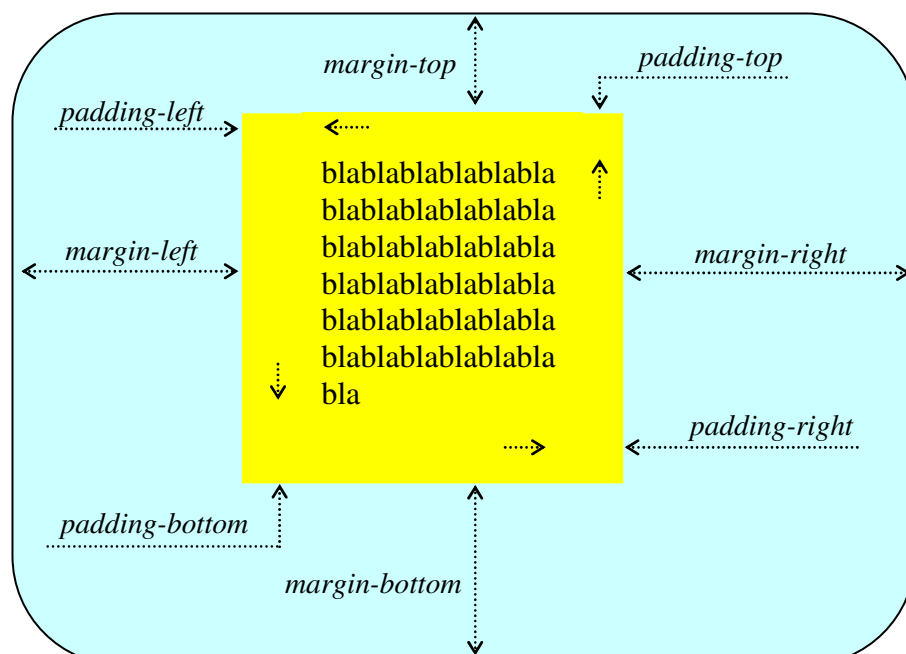
Un bloc se place par défaut en haut à gauche de son conteneur (le body, un autre bloc,...) et occupe toute la largeur de ce conteneur. Ses frères se placeront en-dessous.

**Les marges permettent de positionner un bloc au sein de son parent :**

- Marges externes définies par la propriété *margin*.
- Marges internes définies par la propriété *padding*.

Les définitions des marges externes et internes sont similaires :

- **margin-top** et **padding-top** pour la marge supérieure
- **margin-right** et **padding-right** pour la marge droite
- **margin-bottom** et **padding-bottom** pour la marge inférieure
- **margin-left** et **padding-left** pour la marge gauche





On peut affecter les valeurs des quatre marges de deux façons équivalentes en écrivant :

```
/* notation classique */
```

```
#comment{
  margin-top : 20px ;
  margin-right : 30px ;
  margin-bottom : 30px ;
  margin-left : 20px ;
}
```

ou bien

```
/* notation globale */
```

```
#comment{
  margin: 20px 30px 30px 20px;
}
/* les 4 valeurs sont toujours
dans l'ordre top, right, bottom,
left */
```

En notation globale :

margin : 20px ; affecte la même dimension aux 4 marges soit 20px.

margin : 20px, 30px ; affecte 20px en haut et en bas et 30px à droite et à gauche.

margin : 20px, 30px, 10px ; affecte 20px en haut, 30px à droite et à gauche et 10px en bas.

## Les termes *width* et *height* désignent les dimensions d'un bloc

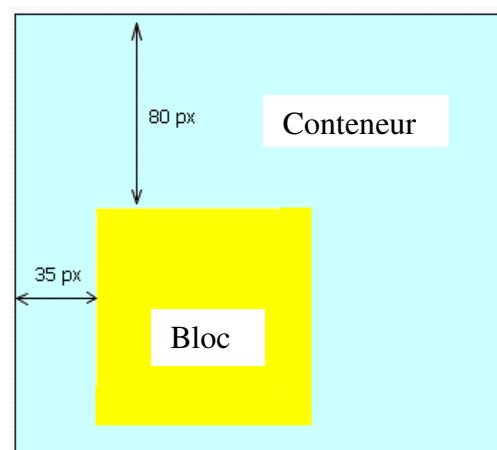
- **width** pour la largeur du bloc
- **height** pour la hauteur du bloc

**Important:** Des marges affectées par défaut et **variables selon les navigateurs**, provoquent des différences de rendu visuel.  
Une solution : affecter aux marges des valeurs choisies.

### Exemples de positionnement

**Placer un bloc jaune de 100px X 100px à 80px du haut du conteneur et à 35px de la gauche du conteneur :**

```
.conteneur {
  padding-top: 80px;
  padding-left: 35px;
}
.bloc {
  width: 100px;
  height: 100px;
  background-color: yellow;
}
```



**Nota :** On emploie ici un *padding-top* et un *padding-left* pour définir le positionnement du bloc à l'intérieur du conteneur (marge interne). Cela signifie que la zone située autour du bloc jaune ne pourra pas être remplie par du texte.

**Deux blocs l'un au-dessous de l'autre avec un espace de séparation :****Partie HTML:**

```
<div class="bloc1">bloc1</div>
<div class="bloc2">bloc2</div>
```

**CSS correspondante:**

```
.bloc1 {
  background-color: aqua;
  height: 50px;
}
.bloc2 {
  background-color: lime;
  height: 50px;
  margin-top: 20px;
}
```

**Résultat :**


bloc1

bloc2

**Trois blocs côte à côte séparés par un espace :****Partie HTML:**

```
<div class="bloc1">bloc1</div>
<div class="bloc2">bloc2</div>
<div class="bloc3">bloc3</div>
```

**CSS correspondante:**

```
.bloc1 {
  background-color: aqua;
  height: 50px;
  width: 100px;
  float: left;
}
.bloc2 {
  background-color: lime;
  height: 50px;
  width: 100px;
  float: left;
  margin-left: 20px;
}
.bloc3 {
  background-color: red;
  height: 50px;
  width: 100px;
  float: left;
  margin-left: 20px;
}
```

**Résultat :**


bloc1

bloc2

bloc3

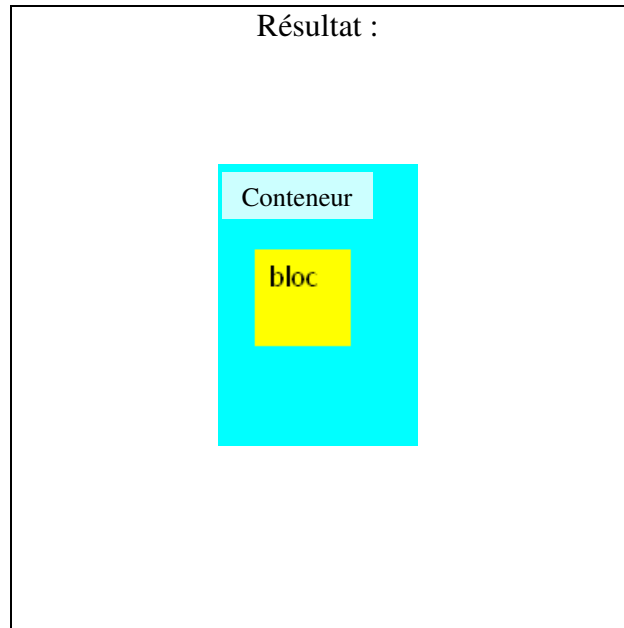
### Un bloc contenu dans un autre bloc:

#### Partie HTML:

```
<div class="conteneur">
  <div class="bloc">bloc</div>
</div>
```

#### CSS correspondante:

```
.conteneur {
  background-color: aqua;
  height: 150px;
  width: 100px;
  padding-top: 40px;
}
.bloc {
  background-color: yellow;
  height: 50px;
  width: 50px;
  margin-left: 20px;
}
```



### 3.2. Positionnement flottant

La propriété **float** permet de positionner un bloc à gauche (**float : left;**) ou à droite (**float : right;**) à l'intérieur du bloc parent (et non-plus l'un en-dessous de l'autre). Le reste du conteneur parent occupera alors l'espace laissé libre, à côté puis éventuellement au-dessous.

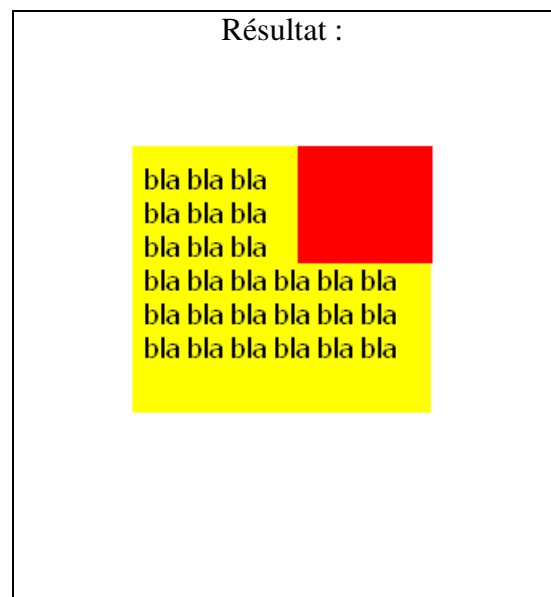
**Exemple :** Une image insérée à droite d'un texte.

#### Partie HTML:

```
<div class="conteneur">
  
  <p>bla bla bla bla bla bla
    bla bla bla bla bla bla
    bla bla bla bla bla...
  </p>
</div>
```

#### CSS correspondante:

```
.conteneur {
  width: 40%;
  background-color: yellow;
}
.image {
  float: right;
}
```



**Attention :**

Le bloc positionné en flottant sort du **flux courant des informations** : sa hauteur n'est donc plus limitée à celle du conteneur. Si aucune hauteur ne lui a été affectée, le bloc peut déborder du conteneur...

Nota : Dans le cas de plusieurs éléments flottants, il est souvent préférable de les placer dans un parent commun.

Il est possible de cumuler les propriétés *float* pour obtenir plusieurs blocs côte à côte.

### 3.3. Positions absolues, fixes et relatives

#### *Position "absolute", position "fixed"*

L'élément est placé à l'aide des propriétés "**top**" et "**left**" par rapport au coin supérieur gauche :

- de son parent si ce dernier est positionné, ou alors du dernier Ancêtre positionné.
- de la page entière (balise *html*) si aucun Ancêtre n'est positionné.

Conséquences :

- La position de l'élément est indépendante de l'emplacement de sa balise à l'intérieur du code source du Parent. L'élément positionné en absolu ne décale pas et ne perturbe pas les autres données, auxquelles il peut cependant se superposer.
- Ce positionnement rend difficile l'adaptation du site aux différentes résolutions d'écrans des visiteurs.

Différence entre "**absolute**" et "**fixed**" :

- "**absolute**" autorise le défilement des données positionnées
- "**fixed**"<sup>2</sup> interdit le défilement des données positionnées

#### *Position "relative"* :

La position "**relative**" d'un élément est dépendante de la position de l'élément qui le précède dans le code HTML. Le code CSS donne la position finale du bloc à l'aide des propriétés "**top**", "**right**", "**bottom**" et "**left**".

Conséquences :

- L'élément positionné est dépendant des éléments frères.
- Il influence les éléments frères.

---

<sup>2</sup> **Attention "fixed"** ne fonctionne ni avec IE6 ni avec IE7...

## 4. Cascade de styles et priorités des styles

Dans le cas où une page web mentionne plusieurs feuilles de styles, les règles CSS sont construites par le navigateur au fur et à mesure de leur lecture, dans l'ordre où elles sont mentionnées. Ainsi lorsque des règles différentes apparaissent pour un même sélecteur, c'est la dernière règle lue qui prime.

### *Ordre de priorité des règles:*

Avec les conteneurs imbriqués le rendu à l'écran peut ne pas être celui attendu... Cela s'explique par les priorités auxquelles les règles CSS obéissent. Plus le sélecteur d'élément précise la cible, plus sa priorité est importante. L'ordre des priorités par ordre décroissant est le suivant :

1. « `<h1 style="color: black">...</h1>` » (un style direct dans la balise Html) vaut 1000 points de priorité.
2. `h1#sommaire {color: black}` vaut 101 points de priorité: 1 pour l'élément, 100 pour l'identifiant.
3. `h1.noir {color: black}` vaut 11 points de priorité : 1 pour l'élément, 10 pour la classe.
4. `menu h1 {color: black}` vaut 2 points de priorité : un point par élément du sélecteur.
5. `h1 {color: black}` vaut 1 point de priorité.

**Un style ne peut jamais modifier un style de priorité plus élevée.**

Exception à la règle : la mention **!important** qui indique au navigateur une priorité absolue.

*Exemple :*

```
p {
  color: maroon !important;
}
```

Pour ajouter des règles spécifiques, il faut **utiliser le moins possible de sélecteurs.**

*Exemple :*

```
<div id="page"> /* Conteneur */
  <p> ...
    <ul>
      <li> <a>...</li>
      ...
    </ul>
  </p>
</div>
```

Il est inutile de spécifier le style des liens `<a>` à partir de la règle `#page p ul li a` la règle `#page a` suffit.

Par contre, si en dehors des listes d'autres liens d'un style différent figurent dans le conteneur « page » il faudra utiliser `div#page li a`.

# Ecriture des pages d'un site

## 1. Matériel nécessaire

Le matériel minimum pour écrire le code HTML et les CSS est un éditeur élémentaire tel que le *Bloc notes* sous Windows ou mieux *Notepad++* (gratuit).

Cependant, l'écriture des feuilles de style sous Windows est grandement facilitée par l'éditeur de CSS *TopStyle* (gratuit en version 3 pour XP, hélas payant en version 4 pour Vista, W7...):

- indentation automatique des définitions de style
- aide à l'écriture des couples propriétés-valeurs
- aide au choix des nuances de couleurs
- vérification de la syntaxe.

Il faudra aussi vérifier le rendu de la page web avec les différents navigateurs. Sous Windows, outre l'inévitable *Internet Explorer* il sera bon d'installer *Firefox* (gratuit).

Par ailleurs, plusieurs modules complémentaires de Firefox sont des assistants précieux pour la fabrication de sites tel *Web Developer* (aide au développement), *Html Validator* (validateur W3C) et *ColorZilla* (outil pipette pour prélever une couleur et récupérer son codage).

Le passage du site en ligne se fera en utilisant un logiciel FTP (File Transfer Protocol) pour copier les fichiers du site ( **.html** , **.css** et les images **.gif** , **.png** , **.jpg**) sur le serveur d'un hébergeur.

Le logiciel *FileZilla* (gratuit) ou tout autre logiciel FTP remplira cette fonction.

## 2. Etapes de la fabrication d'un site

- Définir l'architecture du site (papier, crayon, et ... cogitations).
- Transcrire en HTML, sans aucune indication de mise en forme, le contenu d'une page. Enregistrer la page sur son ordinateur (extension **.html**). La page d'accueil se nomme généralement *index.html*.
- Ecrire les feuilles de style nécessaires à la mise en forme du source HTML. Enregistrer le fichier sur son ordinateur par exemple *habillage.css*.
- Etablir dans le bloc en-tête head du HTML le lien avec le fichier CSS. Dans notre cas :

```
<head>
    .....
    <link rel="stylesheet" href="habillage.css" type="text/css" />
</head>
```

- Essayer la page sur un navigateur en double-cliquant sur *index.html*.
- Transférer l'ensemble des pages du site sur le serveur d'un hébergeur. A l'issue du transfert, le site est consultable en ligne.

## Créer des menus simples

Il n'existe pas de balise spécifique pour désigner un ensemble de liens. La solution la plus fréquente consiste à traiter le menu de navigation comme une liste de liens, en utilisant la balise `<ul>` correspondant aux listes non ordonnées.

Exemple, pour un menu comportant 4 items:

```
<ul id="navigation">
  <li><a href="#" title="aller à la section 1">Section1</a></li>
  <li><a href="#" title="aller à la section 2">Section2</a></li>
  <li><a href="#" title="aller à la section 3">Section3</a></li>
  <li><a href="#" title="aller à la section 4">Section4</a></li>
</ul>
```

L'identifiant placé dans la balise `<ul>` sera nécessaire pour accéder aux différents éléments du menu et pour les modifier dans la feuille de style.

Résultat à l'écran:

- [Section1](#)
- [Section2](#)
- [Section3](#)
- [Section4](#)

### 1. Un menu vertical

On va donner une largeur au menu (200px), supprimer marges et padding par défaut ainsi que la puce des éléments de la liste:

```
#navigation {
  width: 200px;           /* largeur du menu */
  list-style: none;      /* pas de puce */
  margin: 0;            /* pas de marge */
  padding: 0;           /* pas de padding */
}
```

Résultat à l'écran:

[Section1](#)  
[Section2](#)  
[Section3](#)  
[Section4](#)

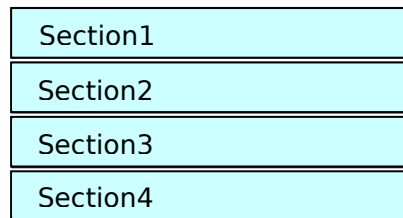
La suppression des marges externes **et** internes est nécessaire pour obtenir le même rendu avec tous les navigateurs : les valeurs par défaut de ces propriétés n'y sont pas identiques.

Le comportement des éléments de liste est de type bloc: ils s'empilent les uns sur les autres et occupent toute la largeur disponible. La disposition des éléments est donc déjà correcte.

On va donner une couleur de fond, une couleur de texte, et une bordure aux éléments <li> ainsi qu'une marge inférieure qui va aérer la liste:

```
#navigation li {
  background: aqua;
  color: black;
  border: 1px solid #600;           /* bordure pour chaque ligne */
  margin-bottom: 1px;           /* marge inférieure " " " */
}
```

Résultat à l'écran:



Pour le fond et le texte il faut veiller au choix de couleurs compatibles et donnant un contraste suffisant.

Pour faire réagir les éléments du menu comme des boutons au survol du pointeur de la souris, les liens doivent occuper tout l'espace à l'intérieur des <li>:

```
#navigation li a {
  display: block;           /* les éléments occupent tout l'espace */
  background: aqua;
  color: black;
  font: 1em "Trebuchet MS", Arial, sans-serif ;
  line-height: 1em ;
  text-align: center ;     /* centrage du texte */
  text-decoration: none ;  /* supprime le soulignement des liens */
  padding: 4px 0 ;        /* padding pour aérer le texte */
}
```

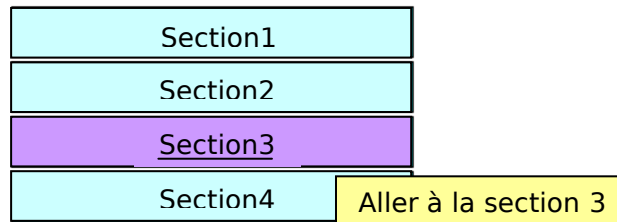
Pour faire en sorte que les liens réagissent au passage de la souris nous emploierons la pseudo-classe *hover*.

**Attention**, IE6 ne gère la pseudo-classe *hover* que lorsqu'elle est appliquée à la balise <a>.

```
#navigation li a:hover {
  background: fuchsia;       /* couleur de fond au survol */
  text-decoration: underline ; /* item souligné au survol */
}
```



Résultat à l'écran:



## 2. Un menu horizontal

Cette fois, il faut modifier le comportement par défaut des `<li>`: ils ne doivent plus s'empiler, mais se placer les uns à côté des autres. Il y a deux approches possibles pour y parvenir.

### Mode de rendu "en-ligne"

On peut modifier le mode d'affichage pour que les `<li>` s'affichent comme des éléments de type "inline":

```
#navigation li {
  display: inline ;
  ...
}
```

Les éléments du menu se comportent maintenant comme du texte "normal".

Résultat à l'écran:

[Section1](#) [Section2](#) [Section3](#) [Section4](#)

Cette méthode permet de réaliser facilement un menu de navigation aligné à droite ou centré, et évite de sortir les `<li>` du flux de la page (on verra plus loin quelques ennuis liés à la sortie de ces éléments du flux). De plus, la largeur des éléments du menu s'adapte automatiquement à la taille du texte.

Malheureusement, en procédant de cette façon, on ne peut plus contrôler les hauteurs et largeurs des `<li>`, on ne peut agir que sur les padding et sur l'interlignage.

Un exemple de menu centré:

```
#navigation {
  margin: 0 ;
  padding: 0 ;
  list-style: none ;
  text-align: center ;
}
```

On supprime les marges, padding et puces de la même manière que pour le menu vertical. La propriété "text-align: center" aura pour effet de centrer *toutes les balises de type inline à l'intérieur de la liste*.

```
#navigation li {
  display: inline ;
  margin-right: 1px ;
  color: #fff ;
  background: #c00 ;
}
```

On modifie le rendu des éléments de liste, et on ajoute couleur de fond, couleur du texte et bordures.

```
#navigation li a {
  padding: 4px 20px ;
  background: #c00 ;
  color: #fff ;
  border: 1px solid #600 ;
  font: 1em "Trebuchet MS", Arial, sans-serif ;
  line-height: 1em ;
  text-align: center ;
  text-decoration: none ;
}
```

**Attention**, il ne faut pas utiliser "display: block" sur les liens, sinon ils vont étirer les éléments de liste sur toute la largeur disponible, les obligeant à se superposer! On peut les agrandir en utilisant les valeurs de padding.

Le comportement réactif des liens est obtenu de la même façon que pour le menu vertical.

```
#navigation li a:hover, {
  background: #900 ;
  text-decoration: underline ;
}
```

## Items flottants

Une autre possibilité est de faire "flotter" les éléments :

```
#navigation li {
  float: left ;
}
```

Dans ce cas, les <li> restent de type bloc, et on peut encore les dimensionner. Par contre, ils sont sortis du flux de la page, et la propriété *text-align* n'aura plus d'effet sur eux. Cette méthode ne sera donc utile qu'avec des éléments de menu *de mêmes dimensions*.

Toujours à partir du même code XHTML, on peut construire un menu de ce type:

```
#navigation {
  margin: 0 ;
  padding: 0 ;
  list-style: 0 ;
}
```

## Suppression des marges, padding et puces (toujours pareil ...)

```
#navigation li {
  float: left ;
  width: 150px ;
  border: 1px solid #600 ;
  margin-right: 1px ;
  color: #fff ;
  background: #c00 ;
}
```

On fait "flotter" les <li> et on leur donne une largeur. Il faut veiller à laisser suffisamment d'espace pour que le texte du lien puisse être agrandi par l'utilisateur! (dans les navigateurs, par appuis successifs de CTRL + pour agrandir, CTRL- pour réduire, CTRL0 pour retrouver la taille nominale).

Le rendu des liens se modifie exactement comme pour le menu vertical:

```
#navigation li a {
  display: block ;
  background: #c00 ;
  color: #fff ;
  font: 1em "Trebuchet MS",Arial,sans-serif ;
  line-height: 1em ;
  padding: 4px 0 ;
  text-align: center ;
}
#navigation li a:hover {
  background: #900 ;
}
```

**Attention :** si on insère du texte dans la page juste en dessous du menu et que celui-ci n'occupe pas toute la largeur, on observe que le texte vient se placer à côté du menu, au lieu de rester sagement en-dessous !

**Explication :**

La propriété *float: left*, a retiré les <li> du flux de courant de la page. Dès lors ils n'entrent plus dans le calcul de la hauteur de la liste <ul>, dont c'est le seul contenu : sa hauteur est donc nulle !!!.

Il existe (au moins) 2 solutions simples à cet inconvénient:

1. appliquer *clear: left* à l'élément qui suit directement le menu dans le code HTML.
2. forcer la liste à occuper une hauteur suffisante :

```
#navigation {
  height: 1.8em ;
  ...
}
```

Mais il n'est pas toujours facile de déterminer la hauteur requise, et des problèmes vont surgir si le visiteur agrandit suffisamment la police de caractères pour obliger le texte des items du menu à passer sur 2 lignes.

## Maîtriser les liens

### Liens internes à une page

Les liens internes permettent au visiteur de se déplacer à l'intérieur d'une page sans être obligé de faire défiler la fenêtre (scrolling). Le visiteur va donc cliquer sur un lien et il sera amené vers la cible de ce lien.

Pour cela deux étapes sont nécessaires:

- **Définir la cible d'un lien (l'ancre)**

Exemple:<sup>3</sup> `<a name="debut"></a>`

Le nom "debut" suffit à caractériser la cible. Il ne doit contenir ni espaces, ni accents, ni caractères spéciaux.

- **Associer un lien à l'ancre définie**

Exemple: `<a href="#debut">Retour au début</a>`

### Liens vers une page différente

C'est l'emploi classique pour assurer la navigation à l'intérieur d'un site. Chaque item du menu est assorti d'un lien vers la page à atteindre.

- **La page est située dans le même répertoire**

Exemple :

`<a href="nom.htm">Cliquez ici</a>`

- **La page est située dans un répertoire différent**

Exemple :

`<a href="sujet/nom.htm">Cliquez ici</a>`

- **Lien vers une ancre d'une autre page et du même répertoire :**

L'arrivée se fait en un endroit précis d'une page différente

Exemple :

`<a href="nom.htm#debut">Cliquez ici</a>`

- **Lien vers une ancre d'une autre page et d'un répertoire différent :**

Cas similaire au précédent mais la page atteinte est dans un autre répertoire.

Exemple :

`<a href="sujet/nom.htm#debut">Cliquez ici</a>`

---

On définit plus commodément la cible d'un lien en utilisant le nom d'élément d'une balise de type bloc :  
Ex : `<div id="article">` ou `<h1 id="titre1">`. (voir *Structure d'une feuille de style*)

## Liens externes au site

La syntaxe de la valeur affectée à l'attribut **href** varie selon le protocole.

- **Lien externe vers un site http**

Exemple :

```
<a href="http://www.site.com">Découvrez SITE</a>
```

- **Lien externe vers un serveur FTP (transfert de fichiers) :**

Exemple :

```
<a href="ftp://ftp.site.com">Transfert de fichiers</a>
```

- **Lien externe vers un fichier :**

Pour faire un téléchargement (fichier .exe ou .zip)

Exemple :

```
<a href="util.exe">Cliquez pour télécharger</a>
```

## Liens e-mail

- La messagerie par défaut du visiteur se lance et l'adresse du destinataire s'inscrit automatiquement (**nom@fournisseur.com** dans les exemples ci-dessous).

Le visiteur reste maître de la dénomination du sujet et du contenu du mail

Exemple :

```
<a href="mailto:nom@fournisseur.com">Responsable</a>
```

- Même chose avec pré-remplissage du sujet du message.

Exemple où le sujet est "**demande**" :

```
<a href="mailto:nom@fournisseur.com?subject=demande">Contact</a>
```

- Même chose avec pré-remplissage du sujet du message et du texte.

Exemple avec le texte "**Blablabla**" :

```
<a href="mailto:nom@fournisseur.com?subject=demande&body=Blablabla">Webmaster du site</a>
```

# Balises HTML

## 1. Les balises bloc

<b>Titre hiérarchique</b>	<p><code>&lt;h1&gt;</code> à <code>&lt;h6&gt;</code></p> <p>Affiche en caractères gras le titre contenu dans la balise. La balise <code>&lt;h1&gt;</code> correspond aux titres de niveau 1. La taille des polices <i>par défaut</i>, affectée automatiquement par le navigateur, diminue quand l'indice augmente.</p> <p>Exemple : <code>&lt;h1&gt;Titre principal&lt;/h1&gt;</code>.</p>
<b>Paragraphe</b>	<code>&lt;p&gt;Texte&lt;/p&gt;</code>
<b>Élément de bloc générique</b>	<p><code>&lt;div&gt;</code> Bloc de texte <code>&lt;/div&gt;</code></p> <p>Balise générique utilisée pour marquer l'unité sémantique d'un texte ou pour sa mise en page par CSS</p>
<b>Filet horizontal</b>	<p><code>&lt;hr /&gt;</code></p> <p>Trace une séparation sur toute la largeur disponible.</p>
<b>Tableau</b>	<pre> &lt;table border="épaisseur"&gt;   &lt;caption&gt; Titre du tableau &lt;/caption&gt;    &lt;tr&gt; &lt;!--1<sup>ère</sup> ligne --&gt;     &lt;th&gt;cellule d'entête&lt;/th&gt;     &lt;th&gt;cellule d'entête&lt;/th&gt;   &lt;/tr&gt;   &lt;tr&gt; &lt;!--2<sup>ème</sup> ligne --&gt;     &lt;td&gt; &lt;!--1<sup>ère</sup> cellule --&gt;       Contenu de la première cellule     &lt;/td&gt;     &lt;td&gt; &lt;!--2<sup>ème</sup> cellule --&gt;       Contenu de la deuxième cellule     &lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt; &lt;!--3<sup>ème</sup> ligne --&gt;     &lt;!--Ici on place de même les cellules     de la 3<sup>ème</sup> ligne du tableau--&gt;   &lt;/tr&gt; &lt;/table&gt; </pre> <p>Quadrillage visible par la mention de <code>border=</code> dans la balise d'ouverture du tableau. "épaisseur" contient l'épaisseur de la bordure extérieure en nombre de pixels</p>
<b>Listes à puces</b>	<pre> &lt;ul&gt;   &lt;li&gt;Texte 1&lt;/li&gt;   &lt;li&gt;Texte 2&lt;/li&gt; &lt;/ul&gt; </pre>
<b>Liste ordonnée</b>	<pre> &lt;ol type="Type"&gt;   &lt;li&gt;Texte 1&lt;/li&gt;   &lt;li&gt;Texte 2&lt;/li&gt; &lt;/ol&gt; </pre> <p>Valeurs données à "Type" :</p> <ul style="list-style-type: none"> <li>a = liste alphabétique minuscules,</li> <li>A = " alphabétique majuscules,</li> <li>i = chiffres romains minuscules,</li> <li>I = chiffres romains majuscules.</li> </ul>

<b>Citation</b>	<code>&lt;blockquote&gt;Texte&lt;/blockquote&gt;</code>
<b>Corps du document</b>	<p><code>&lt;body&gt;</code> Contenu du document <code>&lt;/body&gt;</code></p> <p>Les propriétés CSS affectées à cette balise définissent les caractéristiques générales du document</p>

## 2. Les balises en ligne

<b>Lien hypertexte</b>	<code>&lt;a href="page.html" title="Nos activités"&gt;Page&lt;/a&gt;.</code>
<b>Saut de ligne</b>	<code>&lt;br /&gt;</code>
<b>Renforcement</b>	<code>&lt;strong&gt; &lt;/strong&gt;</code>
<b>Emphase</b>	<code>&lt;em&gt; &lt;/em&gt;</code>
<b>Élément en ligne générique</b>	<p><code>&lt;span&gt;</code> Texte <code>&lt;/span&gt;</code></p> <p>Regroupe au fil du texte des éléments ayant une unité sémantique. Correspond à la balise bloc <code>&lt;div&gt;</code></p>
<b>Insertion d'une image</b>	<p><code>&lt;img src="nom.format" alt="légende"/&gt;</code></p> <p>Tous les navigateurs reconnaissent les formats <b>GIF</b>, <b>JPEG</b> et <b>PNG</b>. Le texte alternatif pour malvoyants ou navigateurs en mode texte est obligatoire en XHTML</p>

# Propriétés CSS

## 1. Marges

Marges externes = **margin** et marges internes = **padding**. La syntaxe est similaire, **mais la valeur *auto* est ignorée de *padding*** :

<b>margin-x</b>	Marge externe en notation classique	<p>Valeurs données à <b>x</b> :</p> <p><b>top</b> = marge supérieure. Valeur numérique, % ou auto.</p> <p><b>right</b> = marge droite. Valeur numérique, % ou auto.</p> <p><b>bottom</b> = marge inférieure. Valeur numérique, % ou auto.</p> <p><b>left</b> = marge gauche. Valeur numérique, % ou auto.</p>
<b>margin</b>	Notation globale	Les 4 valeurs sont toujours sous la forme <b>top right bottom left</b> et toujours dans cet ordre.

**IMPORTANT :** En présence de *padding*, la place occupée par le bloc est égale **aux dimensions augmentées des valeurs des marges internes**.

## 2. Positionnement des blocs

<b>position</b>	Type de positionnement	<p><b>absolute</b> = positionnement mesuré à partir des bords de l'élément parent; peut défiler.</p> <p><b>fixed</b> = positionnement mesuré à partir du bord de l'élément parent; fixe lors du défilement.</p> <p><b>relative</b> = positionnement relatif mesuré à partir de la position normale de l'élément proprement dit.</p> <p><b>static</b> = Dans le flux normal (réglage par défaut).</p>
Emplacement du bloc lorsque <b>position</b> est différent de <b>static</b> .  Autrement, utiliser <b>margin</b> .	placé en haut du conteneur	<b>top</b> : valeur numérique
	placé à droite du conteneur	<b>right</b> : valeur numérique
	placé en bas du conteneur	<b>bottom</b> : valeur numérique
	placé à gauche du conteneur	<b>left</b> : valeur numérique
	définit l'empilement de certains éléments d'une page	<b>z-index</b> : valeur numérique Le bloc au <b>z-index</b> le plus élevée s'affiche en premier plan.
<b>width</b>	Largeur du bloc	valeur numérique ou auto (par défaut)
<b>min-width</b>	Largeur minimale du bloc	valeur numérique



<b>max-width:</b>	Largeur maximale du bloc	valeur numérique. Utiliser avec <code>overflow</code>
<b>height :</b>	Hauteur du bloc	valeur numérique ou <code>auto</code> (par défaut)
<b>min-height:</b>	Hauteur minimale du bloc	valeur numérique
<b>max-height :</b>	Hauteur maximale du bloc	valeur numérique. Utiliser avec <code>overflow</code>
<b>overflow</b>	Action si le contenu dépasse la taille du conteneur	<p><b>visible</b> = Augmentation de la taille du conteneur pour rendre visible tout le contenu</p> <p><b>hidden</b> = Le contenu sera coupé s'il dépasse les limites.</p> <p><b>scroll</b> = Le navigateur propose des barres de défilement.</p> <p><b>auto</b> = Le navigateur décide de la stratégie en cas de dépassement de taille.</p>
<b>float</b>	Positionnement flottant	<p><b>left</b> = L'élément est à gauche du conteneur et entouré par l'élément suivant</p> <p><b>right</b> = L'élément est à droite du conteneur et entouré par l'élément suivant</p> <p><b>none</b> = L'élément ne sera pas entouré (réglage normal).</p>
<b>clear</b>	Annule les effets du positionnement flottant	<p><b>left</b> = poursuite dessous suite à <code>float : left</code>.</p> <p><b>right</b> = la poursuite dessous suite à <code>float : right</code>.</p> <p><b>both</b> = poursuite dessous après <code>float : left</code> et <code>float : right</code>.</p> <p><b>none</b> = pas d'effet (réglage normal).</p>
<b>display</b>	Changement de mode d'affichage	<p><b>block</b> = impose l'affichage en mode bloc - l'élément crée un nouvelle ligne.</p> <p><b>inline</b> = impose l'affichage en ligne</p> <p><b>none</b> = l'élément n'est pas affiché et n'occupe aucun espace.</p> <p><b>flex</b> = modèle de contenus flexibles. En y associant les déclarations:</p> <pre>justify-content: space-around ; flex-wrap: wrap ;</pre> <p>les espaces disponibles autour des blocs sont aménagés et si l'espace n'est pas suffisant on passe à la ligne suivante.</p>
<b>visibility</b>	Affichage avec réservation d'espace	<p><b>hidden</b> = Le contenu de l'élément est caché.</p> <p><b>visible</b> = Le contenu de l'élément est affiché (réglage normal).</p>

### 3. Alignement de texte

<code>text-indent</code>	Retrait de la 1 <sup>ère</sup> ligne	valeur numérique (px, cm, etc...)
<code>line-height</code>	Hauteur de ligne	Si % est utilisé, il est relatif à la taille de la police
<code>vertical-align</code>	Alignement vertical	<b>top</b> aligner en haut, <b>middle</b> aligner au milieu, <b>bottom</b> aligner en bas. <b>sub</b> =mettre en indice (sans réduire la taille de la police). <b>super</b> = mettre en exposant (sans réduire la taille de la police). <b>text-top</b> = aligner sur le bord supérieur de l'espace disponible. <b>text-bottom</b> = aligner sur le bord inférieur de l'espace disponible.
<code>text-align</code>	Alignement horizontal	<b>left</b> (réglage par défaut), <b>center</b> , <b>right</b> , <b>justify</b> (texte justifié).

### 4. Formatage d'écriture

<code>font-size</code>	Taille de la police	<b>xx-small</b> , <b>x-small</b> , <b>small</b> , <b>medium</b> , <b>large</b> , <b>x-large</b> , <b>xx-large</b> , <b>larger</b> , <b>smaller</b> , ou taille définie (px, cm, mm, pt, %)
<code>font-family</code>	type de police	<b>serif</b> , <b>sans-serif</b> , <b>cursive</b> , <b>fantasy</b> , <b>monospace</b> , ou nom de la police ( <b>Arial</b> , <b>Verdana</b> , <b>Times</b> , etc...)
<code>font-weight</code>	Poids de la police	<b>normal</b> , <b>bold</b> , <b>bolder</b> , <b>lighter</b> , ou valeur numérique (de 100 à 900 par tranche de 100)
<code>font-style</code>	style de la police	<b>normal</b> , <b>italic</b> , <b>oblique</b>
<code>font-variant</code>	Variante de la police	<b>normal</b> , <b>small-caps</b>
<code>font</code>	Notation globale des caractéristiques de la police	<b>font-family</b> , <b>font-style</b> , <b>font-variant</b> , <b>font-size</b> , <b>font-weight</b> sont notés par leurs valeurs séparées par des espaces. Exemple : <b>font:Times 13px bold;</b>
<code>color</code>	Couleur du texte	nom standard ou valeur hexadécimale de la couleur
<code>text-decoration</code>	Décoration de texte	<b>underline</b> = souligné. <b>overline</b> = ligne au dessus. <b>line-through</b> = barré. <b>blink</b> = clignotant. <b>none</b> = normal.

<b>text-transform</b>	Casse du texte	<b>capitalize</b> = début des mots en majuscules. <b>uppercase</b> = tout en majuscules. <b>lowercase</b> = tout en minuscules. <b>none</b> = normal.
<b>letter-spacing</b>	Espacement des lettres	<b>normal</b> (par défaut) ou valeur numérique. Les % sont interdits.
<b>word-spacing</b>	Espacement des mots	<b>normal</b> (par défaut) ou valeur numérique. Les % sont interdits.
<b>text-shadow</b>	Lettres avec ombres	<b>h-pos</b> = position horizontale de l'ombre, valeur numérique <b>requis</b> (peut être négative). Les % sont interdits. <b>v-pos</b> = position verticale de l'ombre valeur numérique <b>requis</b> (peut être négative). Les % sont interdits. <b>blur</b> = dégradé de couleur vers le transparent ; valeur numérique <b>optionnelle</b> . Les % sont interdits. <b>color</b> = couleur de l'ombre <b>optionnelle</b> . Par défaut : valeur de la propriété <i>color</i> de l'élément. Exemple : <code>text-shadow 1px 2px 0 #ffff;</code>

## 5. Bordures

<b>border-style</b>	Style de la bordure	<b>none, dotted, dashed, solid, double, groove, ridge, inset, outset.</b>
<b>border-width</b>	Épaisseur de la bordure	<b>thin, medium, thick.</b>
<b>border-color</b>	Couleur de la bordure	Nom ou valeur hexadécimale de la couleur.
<b>border</b>	Notation globale des caractéristiques de la bordure	<b>border-style, border-width</b> et <b>border-color</b> sont notés par leurs valeurs séparées par des espaces. Exemple : <code>border : solid 3px #FFCC99;</code>

## 6. Arrières-plans

<b>background-color</b>	Couleur d'arrière-plan	nom standard ou valeur hexadécimale de la couleur
<b>background-image</b>	Image d'arrière-plan	Cible ou adresse web d'une image gif ou jpeg.
<b>background-repeat</b>	Répétition d'une image	<b>repeat</b> = répéter (réglage par défaut) <b>repeat-x</b> = répéter en "ligne". <b>repeat-y</b> = répéter en "colonne" <b>no-repeat</b> = ne pas répéter, image unique.

<b>background-attachment</b>	Effet de filigrane	<b>scroll</b> = défile avec le texte (réglage par défaut) <b>fixed</b> = l'image d'arrière-plan reste fixe.
<b>background-position</b>	Position d'arrière-plan	<b>top</b> = aligné en haut. <b>center</b> = centrage horizontal. <b>middle</b> = milieu vertical. <b>bottom</b> = aligné en bas. <b>left</b> = aligné à gauche. <b>right</b> = aligné à droite.
<b>background</b>	Notation globale des caractéristiques de l'arrière-plan	<b>background-image</b> , <b>background-repeat</b> , <b>background-attachment</b> et <b>background-position</b> sont notés par leurs valeurs séparées par des espaces. Exemple : <b>background : URL(fond.gif) repeat-x fixed;</b>

## 7. Formatage des listes

<b>list-style-type</b>	Type de représentation	<p>Pour les listes numérotées (ol)</p> <p><b>decimal</b> = numérotation 1., 2., 3., 4. etc...</p> <p><b>lower-roman</b> = numérotation i., ii., iii., iv. etc...</p> <p><b>upper-roman</b> = numérotation I., II., III., IV. etc...</p> <p><b>lower-alpha</b> = numérotation a., b., c., d. etc...</p> <p><b>upper-alpha</b> = numérotation A., B., C., D. etc...</p> <p><b>lower-greek</b> = numérotation grecque: α., β., γ., δ. etc...</p> <p>Listes non numérotées (ul)</p> <p><b>disc</b> = pour les listes ul: : rond plein comme puce.</p> <p><b>circle</b> = pour les listes ul: puce ronde.</p> <p><b>square</b> = pour les listes ul: puce rectangulaire.</p> <p><b>none</b> = pas de puce, pas de numérotation.</p>
<b>list-style-position</b>	Retrait des listes	<b>inside</b> = sans retrait. <b>outside</b> = avec retrait (réglage par défaut).
<b>list-style-image</b>	Puces personnalisées	URL ou chemin d'accès au fichier graphique cible (GIF ou JPEG)
<b>list-style</b>	Notation globale	<b>list-style-type</b> , <b>list-style-position</b> et <b>list-style-image</b> sont notés par leurs valeurs séparées par des espaces. Exemple : <b>list-style: lower-roman inside;</b>

## 8. Formatage des tableaux

<b>caption-side</b>	Position d'un titre (élément HTML <code>caption</code> )	<b>top</b> = en haut du tableau. <b>bottom</b> = sous le tableau. <b>left</b> = à gauche du tableau. <b>right</b> = à droite du tableau.
<b>table-layout</b>	largeurs fixe/variable (élément HTML <code>table</code> )	<b>fixed</b> = Largeur fixe. Le contenu est tronqué s'il ne tient pas dans une cellule. <b>auto</b> = Largeur ajustée au contenu des cellules (par défaut).
<b>border-collapse</b>	Modèle de bordure (élément HTML <code>table</code> )	<b>separate</b> = bordures de cellules de tableau distinctes. <b>collapse</b> = bordures de cellules de tableau confondues.
<b>border-spacing</b>	Espace entre bordures (élément HTML <code>table</code> )	Mention numérique
<b>empty-cells</b>	Affichage de cellules vides (élément HTML <code>table</code> )	<b>show</b> = Les bordures des cellules vides sont affichées. <b>collapse</b> = Les bordures des cellules vides sont cachées (réglage par défaut).

## 9. Les media queries (requêtes de media)

<pre>@media test {   règles   modifiées   applicables   si test est   VRAI }</pre>	<p>Adapter l'affichage des pages au type et à la dimension des périphériques.</p> <p>A utiliser à la fin d'un fichier <b>.css</b></p>	<ol style="list-style-type: none"> <li><i>test</i> contient le type de périphérique ; exemple : <b>print</b> = imprimante. <b>screen</b> = écran. <b>tv</b> = téléviseur.</li> <li><i>test</i> contient une propriété préfixée par <b>min-</b> ou <b>max-</b> et suivie de sa valeur numérique : <b>height</b> = hauteur de l'affichage. <b>width</b> = largeur de l'affichage. <b>device-height</b> = hauteur du périphérique. <b>device-width</b> = largeur du périphérique <i>Le couple propriété-valeur est toujours entre parenthèses.</i></li> <li><i>test</i> contient des conditions combinées par des opérateurs logiques (<b>and</b>, <b>not</b>) ; exemple : (min-width : Xpx) <b>and</b> (max-width: Ypx)</li> </ol>
--	---	--

**media** Associer une feuille de style à un média de destination.

Lien à insérer dans un fichier HTML

*Exemple:*

```
<head>
  ...
  <link rel="stylesheet" media="print"
  href="print.css" type="text/css" />
  ...
</head>
```

## Glossaire Internet

### **Adresse électronique**

Equivalent de l'adresse postale pour la messagerie électronique. S'exprime sous la forme : [nom@fournisseur.domaine](mailto:nom@fournisseur.domaine)

### **Adresse IP**

Adresse affectée à tout équipement informatique connecté à un réseau qui utilise le protocole TCP/IP (cas d'Internet). S'exprime sous la forme : **195 . 154 . 194 . 004** .

### **ADSL**

(*Asynchronous Digital Subscriber Line*). Technologie permettant la transmission de données à haut débit jusqu'à 24Mbit/s sur une ligne téléphonique classique (au lieu du bas débit limité à 56Kbit/s).

- Le débit dans le sens aval (du serveur vers l'ordinateur) est plus élevé que celui dans le sens amont (de l'ordinateur vers le serveur).
- Le débit diminue avec l'augmentation de la distance entre l'utilisateur et le central téléphonique.
- Le téléphone reste libre durant la connexion.

### **Administration**

Aucun ordinateur ne fonctionne tout seul... il faut au minimum assurer les dépannages lorsque cela est nécessaire, surveiller que tout fonctionne correctement, que le disque dur n'est pas plein, que de méchants pirates ne s'attaquent pas aux fichiers, etc... C'est le rôle de l'**administrateur système**. L'**administrateur du serveur web** (la même personne pour les sites de petite taille) se limite à la surveillance du logiciel serveur.

### **Apache**

Ce n'est pas un indien ! C'est le logiciel serveur le plus répandu sur Internet (depuis une bonne dizaine d'années). Logiciel libre, **Apache** a beaucoup contribué à l'essor d'Internet.

### **Balise**

Mots spéciaux du langage HTML qui donnent au navigateur des instructions de mise en forme du texte. Par exemple, pour mettre un texte en gras, il suffit d'écrire :

"<strong>le texte</strong>". **<strong>** est une balise.

### **Base de données**

Une base de données est un programme qui ordonne des données (textes, tableaux de chiffres, photos, ..) de manière structurée (titres d'un côté, articles de l'autre) afin d'y accéder rapidement. Un grand nombre d'hébergeurs proposent la base de données **Mysql**.

### **Blog**

Un blog (contraction de *Binary Log*, littéralement journal numérique) est un site web permettant à une personne d'écrire et mettre en ligne son journal, tout en préservant des fonctions d'interactivité : chacun peut commenter le blog.

Des logiciels spécialisés permettent de s'affranchir de toute contrainte technique.

**Browser Butineur** voir navigateur.

**CMS** (*Content Management System*)

Un CMS (en français *Logiciel de Gestion de Contenu*) est un logiciel permettant de gérer simplement le "contenu" de votre site web (par opposition à la présentation).

SPIP est une sorte de CMS.

**Code, code source**

Désignation d'instructions rédigées en langage informatique connu d'un logiciel. L'écriture d'une page web en langage HTML produit le code reconnu par les navigateurs. On parle de *code source* de la page.

**Courriel, Courrier électronique**

Courriel est le terme officiel, synonyme de courrier électronique.

Le courrier électronique (échange de messages entre personnes, comparable au courrier postal) est une des applications les plus anciennes d'Internet, et sans doute la plus utile. (voir aussi e-mail)

**CSS**

Le langage HTML décrit la structure du texte (titres, paragraphes, etc...). Le langage CSS le complète en décrivant des feuilles de style indiquant la manière d'écrire les titres, les paragraphes et comment effectuer la mise en page.

**DNS** (*Domain Name Server*)

Système d'annuaire distribué sur l'Internet qui donne l'**adresse IP** correspondant à nom de **domaine** (*ttl.fr* par exemple) et réciproquement.

**Easyphp**

C'est un "pack" logiciel sous Windows comprenant trois logiciels libres : **Apache**, **Php** et **Mysql**. L'ensemble de ces trois logiciels vous permet de réaliser une maquette de site web, sur votre PC, que vous pourrez ensuite envoyer sur la machine de votre hébergeur.

**Flash**

Ne souriez pas pour la photo : **Flash** est tout simplement le nom d'un logiciel informatique (dont la licence coûte relativement cher), permettant de réaliser des animations. Attention, l'internaute ne dispose pas obligatoirement de Flash : pensez donc à donner la possibilité de "contourner" le code Flash pour consulter le site.

**Forum web**

C'est comme un grand espace d'affichage dans un lieu public ; chacun peut lire, écrire ce qu'il veut et participer à la conversation... Quand l'espace vient à manquer, les messages les plus anciens sont effacés pour laisser de la place aux nouveaux.

**Fournisseur d'accès**

Comme son nom l'indique, il fournit un accès à Internet, via une ligne téléphonique, le câble, ou tout autre moyen de transmission.

**Ftp**

(*File Transfer Protocol*) Protocole de communication définissant une manière d'échanger des fichiers en utilisant Internet. Un logiciel FTP sera utilisé pour envoyer les pages d'un site web sur un serveur de l'hébergeur.

**Groupe de news**

Imaginez une boîte postale, qui serait mise en place dans le bureau de poste de votre quartier. Vous écrivez un message, vous le mettez dans la boîte, et aussitôt celui-ci est photocopié et envoyé à tous les bureaux de postes de toutes les villes du monde. D'autre part, chaque message reçu (qu'il vienne d'un habitant du quartier ou qu'il ait été envoyé par un autre bureau de poste) est affiché dans le bureau de poste, de sorte que vous pouvez lire tous les messages reçus... jusqu'à ce que le tableau d'affichage soit plein (les employés des PTT retirent alors les messages les plus anciens). Chacun peut donc lire les discussions passant par ce service, et chacun peut y participer.

**Hébergement**

Si un internaute dit qu'il cherche à être hébergé, cela ne veut pas forcément dire qu'il est à la rue mais plutôt qu'il recherche un *serveur* sur lequel il pourrait déposer ses pages html afin que celles-ci soient consultables sur l'Internet.

**Hébergeur**

Société ou organisme chargé de mettre en ligne des sites Internet. Les pages html sont stockées sur des serveurs dont l'hébergeur assure la maintenance, le dépannage et la sécurité à la fois matérielle et logicielle. Les tarifs d'hébergement sont variables selon les exigences de service requises.

**HTML**

(*HyperText Markup Language*). Langage de balisage qui permet une présentation agréable des textes et la navigation dans le texte et en dehors du texte par des liens (hyperliens ou liens hypertexte).

Le html, "langage" au sens informatique du terme, sera interprété par un navigateur.

**Http**

Protocole de communication définissant la manière de transférer des pages html. Lorsqu'une adresse Internet commence par le mot **http://**, cela signifie qu'il s'agit d'une page web.

**Internet**

Littéralement « ce qui relie les réseaux », Internet est le nom donné au **super-réseau** de câbles et de fibres optiques qui permet de relier entre eux des ordinateurs ou d'autres réseaux distants et différents dans le monde entier

Ce réseau universel est la propriété collective de ses utilisateurs. Il permet de relier entre eux divers réseaux humains, réseaux de relations, communautés d'idées. On parle aujourd'hui de communautés virtuelles et de communautés électroniques...

**Internet Explorer**

Logiciel de navigation sur Internet élaboré par Microsoft. Il existe de nombreux autres navigateurs (voir Mozilla, Firefox et Opéra).

**IP (Internet Protocol)**

C'est le protocole standard universel de transmission des données sur le réseau Internet. On parle plus fréquemment de *protocole TCP/IP*.

**javascript**

Javascript est un "langage de scripts" qui s'exécute sur votre PC. Il est surtout utilisé pour donner aux pages web plus de vie ou plus d'interactivité (par exemple en activant des images ou des sons lorsque la souris survole tel ou tel élément).



**Lien hypertexte**

Les liens hypertextes d'une page web apparaissent généralement en lettres bleu souligné. L'utilisateur pointe sa souris sur ce lien, clique, et accède directement à la page de destination indiquée. Cette nouvelle page peut se trouver sur le même serveur, ou être sur une machine située à des milliers de km de là.

**Linux**

Système d'exploitation gratuit, et de ce fait seul vrai rival de Windows .

**Liste de diffusion**

Imaginez un service des PTT au fonctionnement suivant: chaque courrier qui lui est adressé est photocopiée et renvoyée à toutes les personnes figurant sur une liste (les abonnés de cette liste). Très pratique pour la communiquer au sein d'un groupe, le procédé serait assez lourd. Remplacez le terme courrier par courriel, et ce service des PTT par un logiciel et voila le mode de fonctionnement des listes de diffusion.

**Logiciel libre**

Un logiciel libre donne aux termes de sa licence 4 droits fondamentaux à l'utilisateur; ce dernier peut :

- utiliser le logiciel sans aucune restriction (certains logiciels commerciaux vous imposent de payer pour utiliser le logiciel durant une durée limitée).
- analyser le code du logiciel, afin de comprendre comment il marche (et par exemple de vérifier qu'il ne fait pas d'opérations non visibles).
- redistribuer des copies du logiciel (éventuellement, revendre ces copies, par exemple avec un livre qui en explique le fonctionnement).
- modifier le code du logiciel (il est techniquement possible de l'empêcher) et redistribuer ses modifications (ce qui permet d'assurer un développement collectif).

Avec sa démarche créatrice collective, le logiciel libre offre à son utilisateur une réelle liberté et la possibilité d'être acteur des évolutions.

**Mail, e-mail**

voir courrier électronique.

**Modérateur**

Concernant une liste de diffusion, un groupe de news ou un forum web, le modérateur est une personne qui reçoit tous les messages (le pauvre...) et prend la décision de publier ou non chaque message. Il peut éliminer les messages hors sujet ou personnels, tout ce qui nuit à ce genre de débats, et peut amener les lecteurs « sérieux » à s'en détourner.

**Moteur de recherche**

Le nombre de sites sur Internet est extrêmement élevé, la quantité d'information est donc énorme, mais elle n'est absolument pas structurée. Les moteurs de recherche permettent de rechercher les sites consacrés à un thème précis (recherche par mots-clés).

**Mozilla, Firefox**

voir Internet Explorer.

**Multimédia**

Internet est multimedia, car il est possible de faire figurer sur une page html aussi bien du texte que des images ou des sons.

**Mysql**

Voir Base de données

**Navigateur**

Pour surfer sur Internet (passer d'un serveur à l'autre en cliquant sur les liens hypertextes), il faut utiliser un programme informatique : c'est le navigateur (appelé aussi *browser* ou *butineur*).

**Nom de domaine**

Nom apparaissant dans une adresse Internet : *le-pic.org*, *sncf.fr* sont des noms de domaine.

**OpenOffice**

Une suite bureautique, un logiciel libre qui offre quasiment les mêmes fonctionnalités que Microsoft Word, Excel et Powerpoint. En plus, OpenOffice permet de générer très simplement des fichiers au format pdf.

**Opera**

Voir Internet Explorer.

**Page html**

Page écrite en langage html. Chaque page possède une URL (une adresse) qui peut être référencée par un lien hypertexte.

**pdf**

Un format de fichiers, qui garantit la reproduction du document à l'identique sur les écrans de tous formats (contrairement à une page html). La taille d'un document au format pdf est plus réduite que sous les formats issus des traitements de texte : l'affichage sera plus rapide. Enfin, un fichier pdf ne transporte pas de virus et ne peut pas être modifié.

La suite bureautique OpenOffice permet de générer des fichiers pdf de manière très simple

**Php**

Php est un "langage de scripts" qui s'exécute sur le serveur. Il permet de réaliser aussi bien de petits outils pour simplifier la gestion du site, que des logiciels très puissants. SPIP est écrit en Php.

**Pointeur**

voir lien hypertexte.

**Protocole de communication**

Ensemble des règles auxquelles doit obéir tout programme ayant pour objet d'établir une communication.

Sur Internet, ces protocoles sont publics (on dit aussi ouverts) ce qui permet à tout un chacun d'accéder à toutes les informations disponibles mais aussi de réaliser ses propres publications.

**Registrar**

En français : bureau d'enregistrement de nom de domaine. Une société qui s'occupe de réserver et de publier les données de DNS correspondant à votre domaine. En France, Gandi est un de ces bureaux.

**Serveur**

Système informatique (logiciel et machine) destiné à fournir des services à des utilisateurs connectés et, par extension, organisme qui exploite un tel système.

Les pages html sont placées sur des serveurs pour être consultables sur le web. On peut avoir son propre serveur ou choisir d'utiliser les services (gratuits ou non) d'un hébergeur.

### **SPIP**

Un CMS (voir ce mot) simple d'emploi, prisé du monde associatif et du monde professionnel. SPIP est un logiciel libre, qui a besoin pour fonctionner des logiciels **Php** et **Mysql** présents chez la plupart des hébergeurs (**mais pas tous**. Il faudra vérifier si votre hébergeur est dans ce cas).

Pour mettre au point votre site vous pouvez aussi installer SPIP sur votre PC sous Windows (grâce au logiciel **easypHP**), ou sur votre C sous Linux ou votre Mac sous Mac OS X.

### **Squelette**

Dans un site sous SPIP, se dit des fichiers (une vingtaine environ) qui permettent de définir la mise en pages du site, les facilités de navigation, la disposition des rubriques, etc...

Un fichier CSS est toujours associé au squelette, il permet en particulier de sélectionner les polices, et les couleurs qui vont habiller le squelette.

### **Syndication**

Deux sites web syndiqués fonctionnent en tandem : les mises à jour de l'un sont automatiquement reflétées par l'autre.

Grâce à la syndication, une association qui possède plusieurs sites régionaux, peut obtenir d'afficher en région, sans intervention des webmestres, des informations nationales parfaitement à jour.

### **Tag**

voir balise.

### **TCP/IP**

voir IP

### **Téléchargement**

Transfert de programmes ou de données d'un ordinateur vers un autre. Pour un internaute, le téléchargement est le plus souvent un transfert vers son propre ordinateur depuis un serveur (sens aval), ou parfois depuis l'ordinateur vers le serveur (sens amont).

### **URL**

Adresse d'une page html par exemple, ou de tout autre objet (par exemple, une adresse électronique est une URL).

### **Validation html ou Xhtml**

Si vous voulez que votre page html soit correctement représentée sur tous les navigateurs (et pas seulement sur le vôtre), il est important qu'elle soit validée "conforme aux spécifications" du langage html. Dans le cas contraire, vous laissez le champ libre au navigateur, qui interprètera le code à sa façon... et les différents navigateurs n'ont pas la même manière !

### **Web**

En anglais web, est l'abrégié de *World Wide Web* (toile d'araignée mondiale). La plupart des adresses Internet commencent par les initiales de ce mot : www.

La multitude des liens hypertextes qui permettent, d'un clic de souris, de passer d'une page html à l'autre évoque les fils de la toile d'araignée.

**Webmaster, webmestre**

Personne qui assure la maintenance d'un site Internet. L'adresse de courriel du webmestre figure le plus souvent sur les pages d'un site pour servir de contact aux internautes qui ont des demandes, des observations ou des suggestions à faire.

**W3C**

(*World Web Consortium*). Organisme qui définit les normes des langages HTML et CSS.

**Wiki**

Un wiki est un site web que l'on construit "à plusieurs" : toutes les personnes ayant accès au wiki (c'est-à-dire, bien souvent tout le monde) peuvent écrire des articles, et ainsi faire vivre et grossir le site. Techniquement, Wikipedia est construit autour d'un wiki.

**Wikipedia**

Une encyclopédie libre, ce qui signifie que chacun peut participer à l'élaboration de cette encyclopédie, et ainsi partager ses connaissances.

**WYSIWYG**

Acronyme de *What You See Is What You Get*. Un éditeur HTML est de type WYSIWYG lorsque le source html reflète exactement ce qui apparaît à l'écran de composition de la page. Ces logiciels facilitent la création de pages web (pas de nécessité de connaître le langage HTML) mais produisent souvent un code de qualité médiocre.

**Xhtml**

(*eXtended HTML*) Désigne le langage **HTML étendu** (voir Html), conforme aux dernières spécifications du W3C.

## Codification des caractères spéciaux

<i>Caractère</i>	<i>Code Iso</i>	<i>Entité</i>	<i>Caractère</i>	<i>Code Iso</i>	<i>Entité</i>
À	&#192;	&Agrave;	à	&#224;	&agrave;
Á	&#193;	&Aacute;	á	&#225;	&aacute;
Â	&#194;	&Acirc;	â	&#226;	&acirc;
Ã	&#195;	&Atilde;	ã	&#227;	&atilde;
Ä	&#196;	&Auml;	ä	&#228;	&auml;
Å	&#197;	&Aring;	å	&#229;	&aring;
Æ	&#198;	&Aelig;	æ	&#230;	&aelig;
Ç	&#199;	&Ccedil;	ç	&#231;	&ccedil;
È	&#200;	&Egrave;	è	&#232;	&egrave;
É	&#201;	&Eacute;	é	&#233;	&eacute;
Ê	&#202;	&Ecirc;	ê	&#234;	&ecirc;
Ë	&#203;	&Euml;	ë	&#235;	&euml;
Ì	&#204;	&Igrave;	ì	&#236;	&igrave;
Í	&#205;	&Iacute;	í	&#237;	&iacute;
Î	&#206;	&Icirc;	î	&#238;	&icirc;
Ï	&#207;	&Iuml;	ï	&#239;	&iuml;
Ð	&#208;	&ETH;	ð	&#240;	&eth;
Ñ	&#209;	&Ntilde;	ñ	&#241;	&ntilde;
Ò	&#210;	&Ograve;	ò	&#242;	&ograve;
Ó	&#211;	&Oacute;	ó	&#243;	&oacute;
Ô	&#212;	&Ocirc;	ô	&#244;	&ocirc;
Õ	&#213;	&Otilde;	õ	&#245;	&otilde;
Ö	&#214;	&Ouml;	ö	&#246;	&ouml;
×	&#215;	&times;	÷	&#247;	&divide;
Ø	&#216;	&Oslash;	ø	&#248;	&oslash;
Ù	&#217;	&Ugrave;	ù	&#249;	&ugrave;
Ú	&#218;	&Uacute;	ú	&#250;	&uacute;
Û	&#219;	&Ucirc;	û	&#251;	&ucirc;
Ü	&#220;	&Uuml;	ü	&#252;	&uuml;
Ý	&#221;	&Yacute;	ý	&#253;	&yacute;
Þ	&#222;	&THORN;	þ	&#254;	&thorn;

## Codification des caractères spéciaux

<b>ß</b>	&#223;	&szlig;	ÿ	&#255;	&yuml;
?	&#129;		ı	&#161;	&iexcl;
,	&#130;		¢	&#162;	&cent;
f	&#131;		£	&#163;	&pound;
„	&#132;		¤	&#164;	&curren;
...	&#133;		¥	&#165;	&yen;
†	&#134;		ı	&#166;	&brvbar;
‡	&#135;		§	&#167;	&sect;
^	&#136;		¨	&#168;	&uml;
%o	&#137;		©	&#169;	&copy;
Š	&#138;		ª	&#170;	&ordf;
<	&#139;		«	&#171;	&laquo;
<b>Œ</b>	&#140;	&Oelig;	¬	&#172;	&not;
□	&#141;			&#173;	&shy;
Ž	&#142;		®	&#174;	&reg;
□	&#143;		ˉ	&#175;	&masr;
□	&#144;		°	&#176;	&deg;
‘	&#145;		±	&#177;	&plusmn;
’	&#146;		²	&#178;	&sup2;
“	&#147;		³	&#179;	&sup3;
”	&#148;		´	&#180;	&acute;
•	&#149;		µ	&#181;	&micro;
–	&#150;		¶	&#182;	&para;
—	&#151;		•	&#183;	&middot;
~	&#152;		¸	&#184;	&cedil;
™	&#153;		¹	&#185;	&supl;
š	&#154;		º	&#186;	&ordm;
>	&#155;		»	&#187;	&raquo;
<b>œ</b>	&#156;	&oelig;	¼	&#188;	&frac14;
□	&#157;		½	&#189;	&frac12;
ž	&#158;		¾	&#190;	&frac34;
ÿ	&#159;		ı	&#191;	&iquest;
<b>Espace</b>	&#160;	&nbsp;	€	&#8364;	&euro;

Ces feuillets sont largement inspirés :

- de plusieurs sites consacrés au sujet :
  - <http://www.alsacreations.com/> tutoriels CSS.
  - <http://fr.selfhtml.org> références du HTML et des CSS.
- de l'excellent ouvrage de Raphaël GOETTER *CSS2 pratique du design web*