

# HTML - XHTML

# SOMMAIRE :

<b>QU'EST-CE QUE HTML ? .....</b>	<b>5</b>
HISTOIRE DU HTML .....	5
VERSIONS DU HTML .....	5
VERSIONS DU XHTML .....	6
PAGES STATIQUES VS PAGES DYNAMIQUES .....	6
<b>APPRENDRE LE HTML .....</b>	<b>7</b>
HTML ET XHTML – NORMES D'ECRITURE .....	7
A SAVOIR : .....	8
<b>LA PAGE DE BASE .....</b>	<b>9</b>
LE PROBLEME DES ACCENTS .....	10
<b>LES BALISES META .....</b>	<b>12</b>
META DESCRIPTION .....	12
META KEYWORDS .....	12
META LANGUAGE .....	12
META ROBOTS .....	13
META REVISIT-AFTER .....	13
META PRAGMA .....	13
META EXPIRES .....	13
META REFRESH .....	13
<b>LE CONTENU DE LA PAGE – LE TEXTE .....</b>	<b>14</b>
<b>LES TITRES .....</b>	<b>14</b>
LE TEXTE .....	14
LE RETOUR A LA LIGNE .....	14
<b>QUELQUES STYLES DE TEXTE .....</b>	<b>15</b>
GRAS .....	15
ITALIQUE .....	15
GRAS ET ITALIQUE .....	15
EXPOSANT .....	15
INDICE .....	15
SOULIGNE .....	15
BARRE .....	15

<b>LES LISTES .....</b>	<b>16</b>
LES LISTES NON NUMEROTEES (SIMPLES) .....	16
LES LISTES NUMEROTEES .....	16
LISTES DE DEFINITION .....	16
<b>LES LIENS .....</b>	<b>17</b>
LES LIENS RELATIFS .....	17
LES LIENS ABSOLUS .....	18
FAIRE UN LIEN QUI S'OUVRE DANS UNE NOUVELLE FENETRE .....	18
LE CHOIX DES NOMS DE FICHIERS ET DE DOSSIERS .....	18
FAIRE UN LIEN VERS UN EMAIL (LIEN MAILTO) .....	19
FAIRE UNE ANCRE .....	19
<b>LES IMAGES.....</b>	<b>20</b>
LES IMAGES SUR LE WEB .....	20
LES IMAGES : AFFICHER LES IMAGES .....	21
L' ATTRIBUT : ALT (TEXTE ALTERNATIF).....	21
LES TYPES D'IMAGE .....	22
LES IMAGES-MAP .....	23
<b>LES TABLEUX.....</b>	<b>24</b>
ATTRIBUT : ESPACEMENT DE CELLULES (CELLSPACING) .....	25
ATTRIBUT : LE REMPLISSAGE DANS LES CELLULES (CELLPADDING) .....	26
<b>LES FORMULAIRES .....</b>	<b>27</b>
<b>ELEMENTS DE FORMULAIRE .....</b>	<b>27</b>
LE CHAMP TEXTE NORMAL .....	27
LES BOUTONS RADIO.....	27
LES CASES A COCHER .....	27
LA LISTE DEROULANTE .....	28
LA BOITE DE DIALOGUE .....	28
LES BOUTONS .....	28
LES CHAMPS CACHES .....	28
LE CHAMP PASSWORD.....	28
LES LABELS.....	29
<b>LES CADRES .....</b>	<b>30</b>
LE CONTENEUR <FRAMESET> .....	30

<b>ATTRIBUTS DU CONTENEUR &lt;FRAMESET&gt;</b> .....	<b>31</b>
LES ATTRIBUTS ROWS ET COLS :	31
L'ATTRIBUT FRAMEBORDER.....	31
L'ATTRIBUT BORDER .....	31
L'ATTRIBUT BORDERCOLOR.....	32
L'ATTRIBUT FRAMESPACING .....	32
LA BALISE <FRAME> .....	32
L'ATTRIBUT SRC .....	32
L'ATTRIBUT NAME .....	32
LES ATTRIBUTS MARGINWIDTH ET MARGINHEIGHT .....	32
L'ATTRIBUT FRAMEBORDER.....	33
L'ATTRIBUT BORDER (NETSCAPE UNIQUEMENT) .....	33
L'ATTRIBUT NORESIZE .....	33
L'ATTRIBUT SCROLLING.....	33
LES NAVIGATEURS NON COMPATIBLES .....	34
<b>CADRES LOCAUX &lt;IFRAME&gt;</b> .....	<b>35</b>
LIENS VERS D'AUTRES FRAMES.....	36
<b>ANNEXE 1 : CARACTERES SPECIAUX HTML</b> .....	<b>37</b>
CARACTERES GENERAUX.....	37
MONNAIES .....	39
CARACTERES ALPHABETIQUES ACCENTUES ET SPECIAUX .....	39
ALPHABET GREC.....	41
SCIENCES .....	42
CARACTERES DIVERS : .....	43
<i>Reconnus sous IE6.0+ avec une police unicode</i> .....	43
<i>Reconnus sous IE6.1+ avec une police unicode</i> .....	44
<b>ANNEXE 2 : LES COULEURS HTML "WEB SAFE"</b> .....	<b>45</b>
<b>ANNEXE 3 : LES NOMS DES COULEURS HTML</b> .....	<b>46</b>
LIENS : .....	47
LIVRES :.....	47

# Qu'est-ce que HTML ?

Le HTML (HyperText Markup Language) est le langage de publication utilisé sur le World Wide Web.

XHTML est une simple reformulation de HTML 4, il ne s'agit donc que d'un changement de syntaxe.

## HISTOIRE DU HTML

C'est Tim Berners-Lee qui l'a écrit pour répondre aux besoins du système qu'il venait d'inventer au Cern : le World Wide Web. Ce langage a été conçu pour fonctionner avec le protocole HTTP. Moyen simple et efficace d'accéder sur un réseau à des documents liés entre eux par des hyperliens. Les documents s'enchaînent les uns les autres par un simple clic de souris : écrits en HTML, ils sont transportés jusqu'au lecteur grâce au protocole HTTP

## VERSIONS DU HTML

### *HTML 1.0 :*

Le langage HTML est né en 1991 au CERN (Centre d'études et de recherches nucléaires) situé près de Genève. Il y avait peu de commandes : on pouvait cependant insérer des images, créer des liens hypertextes, mais tout cela en noir, sur un fond gris, et sans interface graphique !

### *HTML 2.0 :*

Le standard HTML 2.0 fut publié en 1995. Avec cette norme, on a vu apparaître des possibilités de mise en page avec des tableaux.

### *HTML 3.0 :*

En Mai 1996, le HTML 3.0 apparaît avec de nouveaux attributs et de nouvelles commandes.

### *HTML 4.0 :*

Publié officiellement au début de l'année 1998 fournit surtout, outre de nouvelles commandes, les feuilles de style en cascade (css).

### *HTML 5 :*

Le futur du web. Malgré le fait que sa spécification complète ne devrait être terminée qu'au mieux fin 2010, les grands acteurs se préparent déjà à son arrivée.

## VERSIONS DU XHTML

XHTML 1.0 strict

XHTML 1.1

Selon le W3C, le XHTML est une reformulation de HTML en XML (<http://fr.wikipedia.org/wiki/XML>).

Il s'agit donc de XML qui va se comporter comme du HTML. C'est le langage standard de balisage actuel, qui remplace le HTML4. Il est conçu pour bien s'interfacer avec les langages à balises, protocoles et applications fondés sur le XML, existants ou futurs.

Il est plus cohérent que le HTML (toutes les balises doivent être fermées). Les anciens navigateurs n'ont pas de problèmes pour lire le XHTML, les nouveaux navigateurs le préfèrent au HTML et certains même au HTML4. Il fonctionne aussi dans les périphériques sans fil, les lecteurs d'écran.

Il appartient à une famille de standards du Web permettant de contrôler l'apparence des pages web sur plusieurs plateformes et navigateurs.

Il permet de faire tester vos pages dans les services de validation du balisage.

Le HTML est un langage de description qui utilise des balises ou "tags" en anglais.

Les balises sont des séquences de caractère, interprétées par les navigateurs Web. Lorsque l'on regarde le contenu d'une page HTML, on repère assez facilement les balises HTML : ce sont les portions de texte semblables à celle-ci : `<strong>ceci est du gras</strong>` Une balise HTML commence par un "<" et se termine par un ">". En html, la plupart des balises doivent être fermées. La majorité des balises servent à la mise en page (texte en gras, les tableaux...), mais toutes ne sont pas dédiées à la mise en page des documents. En effet, le HTML permet par exemple de donner des informations sur le document lui-même (informations qui ne seront pas affichées), d'appeler des fichiers externes comme des images, des animations Flash, des sons...

Les retours à la ligne dans le code, de même que les sauts de ligne et les tabulations ne sont (en général) pas importants.

## PAGES STATIQUES VS PAGES DYNAMIQUES

Les pages que vous verrez sur le Web qui ont une extension autre que ".html" ou ".htm" comme ".php" par exemple sont des pages HTML construites de façon dynamique car contenant des résultats de requêtes, ou de calculs ... Le langage de script (PHP par exemple) s'inclut dans le code HTML mais est interprété par le serveur avant que la page HTML ne soit affichée. Alors que le HTML, comme le Javascript, est interprété par le client (le navigateur). Quoi qu'il arrive, que l'extension de la page soit ".html" ou autre chose, le résultat final sera le même : du code HTML interprété par le navigateur.

# Apprendre le html

Apprendre le HTML ne nécessite pas de connaître de langage de programmation. Il est toujours possible de voir comment les pages sont faites. Connectez-vous sur une page avec votre navigateur favori, et regardez le fichier "source" (Afficher la source, ou avec le clic droit de la souris lorsque le pointeur est sur la page). On peut écrire du HTML avec un éditeur de texte de base (Notepad, Notepad++, ou TextEdit, TextMate, TextWrangler), ou en utilisant un traitement de texte à condition d'enregistrer le fichier au format "texte seulement", avec l'extension html. On peut également utiliser des éditeurs HTML qui génèrent le code. Il est toutefois indispensable de connaître le langage pour utiliser correctement ces éditeurs et profiter ensuite pleinement des CSS.

## HTML ET XHTML – NORMES D'ECRITURE

En xhtml, certaines normes d'écriture sont importantes :

### Bas de casse obligatoirement

Tous les éléments doivent être en bas de casse : `<p> ...</p>` `<img src ... />` Attention si vous utilisez des éditeurs WYSWYG, pour les instructions de javascript, OnMouseOver risque de poser problème. Il faut en effet écrire : `onmouseover`

### Valeurs des attributs entre guillemets et séparés par des espaces

```

```

### Chaque attribut appelle une valeur

Dans le cas de "nowrap" ou de checked par exemple, il faudra donc écrire

```
<td nowrap="nowrap"><input type="checkbox" name="cinema" value="scifi" checked="checked" /></td>
```

### Refermez toutes les balises

`<p>` ou `<li>` n'étaient pas obligatoirement refermées en HTML. EN XHTML c'est obligatoire.

```
<p>Mon paragraphe de texte</p>
<ul>
  <li>menu 1</li>
  <li>menu 2</li>
</ul>
```

### Fermez les balises vides

Les balises qui ne pouvaient pas se fermer en HTML, comme `<img>`, `<br>` se ferment au moyen d'un slash à la fin, précédé d'un espace :

```
<br />
```

### Pas de tirets à l'intérieur des commentaires

Les commentaires en HTML commencent par "<!--" et se terminent par "-->", mais entre ces 2 repères il n'est plus possible, en XHTML, de mettre une autre série de tirets.

```
<!-- mon commentaire et ----- un autre --> n'est donc plus possible.
Ils peuvent par contre être remplacés par des signes "=" par exemple :
<!-- mon commentaire et ===== un autre -->
```

### A SAVOIR :

Les retours à la ligne se codent : <br /> qui signifie "(line)break".

Le paragraphe <p> ...</p> fait un retour à la ligne et un saut de ligne automatiquement.

Comme on l'a vu juste au dessus, on peut mettre des commentaires dans les pages html de la façon suivante :

```
<!-- ceci est un commentaire -->
```

Tout ce qui se trouve entre les tirets sera ignoré par les navigateurs. Ce sera bien présent dans le code, mais ne sera pas interprété et donc pas affiché.

Enfin, les balises ne peuvent pas se chevaucher.

```
<p><strong>Texte en gras</strong></p> → correct
<p><strong>Texte en gras</p></strong> → incorrect
```

# La page de base

Elle est divisée en deux parties :

Première partie : l'en-tête

```
<html>
  <head>
    <title> Le titre de la page</title>
  </head>
```

Deuxième partie : le corps de la page

```
  <body>
    ce qui apparaît dans la
    fenêtre du navigateur
  </body>
</html>
```

Dans l'en-tête viendront d'autres balises :

- les `<meta>`
- les descriptions des styles css
- du Javascript
- la balise `<title>`

Pour cette dernière, il est recommandé soit 7 ou 8 mots maxi, soit entre 50 et 100 caractères selon la tolérance les moteurs de recherches.

Elle sera utilisée par les moteurs de recherches dans le cadre de la recherche ET de l'affichage de la page de résultats, il est donc important d'y insérer des mots-clefs.

Elle est utilisée dans l'historique du navigateur : éviter donc de mettre le même titre dans chaque page

Elle est la première chose que les malvoyants entendent lorsque leur lecteur d'écran lit une page, donc indispensable pour l'accessibilité.

Les documents XHTML doivent commencer par indiquer aux navigateurs comment les interpréter et aux services de validation comment tester leur conformité. C'est une déclaration de type de document (DOCTYPE). DOCTYPE est toujours écrit en majuscules.

Il existe trois DTD :

- Transitional : un peu rigide
- Strict : rigide
- Frameset : qui permet d'inclure des cadres.

Le DOCTYPE doit être placé tout en haut du document avant les éléments <head>, <title> et <meta>

### Les doctypes HTML :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
```

### Les doctypes XHTML :

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

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

<!DOCTYPEhtml PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

### LE PROBLEME DES ACCENTS

Selon le type d'encodage choisi, il faudra ou non coder les accents. L'encodage se définit dans une balise META dans l'en-tête de la page, de la façon suivante :

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Ici on a défini l'encodage "UTF-8" qui est l'encodage actuellement recommandé.

Il fut un temps où l'encodage "ISO8859-1" (aussi appelé Latin-1) était le plus répandu. On le trouve encore comme réglage par défaut sur certains logiciels.

Avec l'encodage ISO8859-1, il est nécessaire de convertir les caractères accentués en leur entité HTML. Quelques exemples sont regroupés dans le tableau ci-dessous :

é	&eacute;	Espace insécable	&nbsp;
è	&egrave;	€	&euro;
à	&agrave;	£	&pound;
ô	&ocirc;	¥	&yen;
â	&acirc;	©	&copy;
ï	&iuml;	®	&reg;
ü	&uuml;	™	&trade;
œ	&oelig;	°	&deg;
ç	&ccedil;	«	&laquo;
õ	&otilde;	»	&raquo;

Vous trouverez plus de codes HTML dans l'Annexe 1. De plus les éditeurs HTML WYSIWYG (type DreamWeaver) s'occupent de coder directement ces caractères, ou du moins une grande majorité, directement.

Avec l'encodage utf-8, l'utilisation de ces codes n'est plus nécessaire. C'est une des raisons de l'utilisation de plus en plus massive de cet encodage.

# Les balises META

Les balises META, placées dans la tête du document HTML (entre les balises <head> et </head>), renseignent les moteurs de recherches sur la nature et le contenu de la page.

Il en existe de nombreuses. Certaines sont très utiles voire indispensables, d'autres beaucoup moins.

## META DESCRIPTION

*Conseils* : Limiter cette balise à 20 / 25 mots. Construire des phrases cohérentes. Concentrer l'essentiel des mots clefs dans cette balise.

Google affiche volontiers le contenu de cette balise dans la description du site sur la page de résultats.

```
<meta name="description" content="Le texte de description de votre page." />
```

Taille de la balise limitée en nombre de mots (20 à 50) ou en nombre de caractères (entre 180 et 300 selon les moteurs de recherches). Les majuscules, caractères accentués et chiffres sont acceptés.

## META KEYWORDS

*Conseils* :

- Limiter la taille des mots-clefs à 1000 caractères.
- Ne répétez pas le même mot-clef plusieurs fois (ça peut être considéré comme du spam par les moteurs de recherches et donc limiter le référencement du site.
- N'utilisez pas le nom de vos concurrents (c'est très mal vu aussi ...).

Google ne tient pas uniquement compte de cette balise meta pour les recherches. Pour lui, le plus important est le contenu de la page.

```
<meta name="keywords" content="motclef1, motclef2, motclef3" />
```

Séparateur conseillé : virgule + espace  
Taille de la balise : de 100 caractères à + de 1800 caractères suivant les moteurs. Les caractères accentués sont acceptés.

## META LANGUAGE

Permet de définir quel est le langage de la page.

```
<meta http-equiv="content-language" content="fr" />
```

## META ROBOTS

Indique au robot s'il doit indexer la page (index ou noindex), suivre les liens (follow ou nofollow), indexer toutes les pages (all)

L'idéal est de créer un fichier robots.txt, et de le placer à la racine du site (au même niveau que la page index). Ce fichier permet d'interdire certains répertoires aux robots (cgi, stats, espaces privés, etc).

Exemple de syntaxe d'un fichier robot standard :

```
User-agent: * (l'étoile désigne l'ensemble des robots)
Disallow: /cgi-bin/ (cette ligne interdit l'indexation du répertoire cgi-bin)
Disallow: /logs/ (cette ligne interdit l'indexation du répertoire logs)
```

```
<meta name="robots" content="Index,NoFollow" />
```

## META REVISIT-AFTER

Demande au robot du moteur de recherches de repasser sur le site après un délai convenu. Valeurs possibles : Numérique + Argument days weeks, ou month

```
<meta name="revisit-after" content="30days" />
```

## META PRAGMA

Impose au navigateur de votre visiteur de recharger la page à chaque nouvelle visite, et donc de voir les dernières mises à jour de votre site. C'est très pratique en phase de développement car le cache du navigateur peut parfois jouer des tours.

```
<meta http-equiv="pragma" content="no-cache" />
```

## META EXPIRES

Donne la date (jj, mois anglais, aaaa) de fin de validité d'une page.

```
<meta http-equiv="expires" content="unodate " />
```

## META REFRESH

Permet de faire recharger automatiquement la page au bout d'un certain temps (en secondes). Elle permet aussi de rediriger vers une autre page automatiquement.

```
<meta http-equiv="refresh" content="10 " /> Rechargement au bout de 10 secondes.
<meta http-equiv="refresh" content="2,url=http://www.google.fr" /> Redirection vers Google au bout de 2 secondes
```

Ne pas hésiter à regarder les sources des sites ou à chercher sur le net pour en découvrir de nombreuses autres.

# Le contenu de la page – Le texte

## LES TITRES

Certaines balises permettent de définir des titres. Elles sont au nombre de 6 et servent à faire le sommaire d'une page. Elles génèrent automatiquement un retour à la ligne et un saut de ligne. Les tailles sont prédéfinies et ces balises apparaissent en gras.

<code>&lt;h1&gt;titre de niveau 1&lt;/h1&gt;</code>	<b>titre de niveau 1</b>
<code>&lt;h2&gt;titre de niveau 2&lt;/h2&gt;</code>	<b>titre de niveau 2</b>
<code>&lt;h3&gt;titre de niveau 3&lt;/h3&gt;</code>	<b>titre de niveau 3</b>
<code>&lt;h4&gt;titre de niveau 4&lt;/h4&gt;</code>	<b>titre de niveau 4</b>
<code>&lt;h5&gt;titre de niveau 5&lt;/h5&gt;</code>	<b>titre de niveau 5</b>
<code>&lt;h6&gt;titre de niveau 6&lt;/h6&gt;</code>	<b>titre de niveau 6</b>

La balise h1 est très importante pour les moteurs de recherches car c'est celle qui a la plus grande importance et qui donc doit normalement identifier le contenu de la page.

## LE TEXTE

Les paragraphes de texte sont contenus dans des balises `<p></p>`

```
<p>Mon paragraphe de texte</p>
```

Par défaut, cette balise génère automatiquement un saut de ligne entre 2 paragraphes.

## LE RETOUR A LA LIGNE

Pour faire un simple retour à la ligne sans créer de nouveau paragraphe (et donc sans saut de ligne), il faut utiliser la balise `<br />`. Celle-ci s'insère dans le code, dans le courant du texte.

```
<p>Mon pragraphe<br />de texte</p>
```

## Quelques styles de texte

### GRAS

Pour mettre en gras, on utilise la balise `<strong>`

```
<strong>texte en gras</strong>
```

### ITALIQUE

Pour mettre en italique, on utilise la balise `<em>`

```
<em>texte en italique</em>
```

### GRAS ET ITALIQUE

Pour mettre en gras ET en italique, on utilise les 2 balises :

```
<strong><em>texte en gras et en italique</em></strong>
```

ou bien aussi :

```
<em><strong >texte en gras et en italique</strong ></em>
```

L'ordre n'a ici pas d'importance.

### EXPOSANT

Pour mettre en exposant, on utilise la balise `<sup>`

```
20<sup>e</sup> siècle
```

### INDICE

Pour mettre en indice, on utilise la balise `<sub>`

```
Molécule d'eau : H<sub>2</sub>O
```

### SOULIGNE

Pour souligner, on utilise la balise `<u>`

```
<u>Texte souligné</u>
```

### BARRE

Pour barrer un texte, on utilise la balise `<del>`

```
<del>Texte barré</del>
```

# Les listes

## LES LISTES NON NUMEROTEES (SIMPLES)

```
<ul>
  <li>premier élément</li>
  <li>deuxième élément</li>
</ul>
```

- premier élément
- deuxième élément

<ul> signifie : "unordered list",

<li> signifie "list item" et fait par défaut apparaître la puce.

## LES LISTES NUMEROTEES

```
<ol>
  <li>premier élément</li>
  <li>deuxième élément</li>
</ol>
```

1. premier élément
2. deuxième élément

<ol> signifie : "ordered list",

<li> fait apparaître le chiffre.

## LISTES DE DEFINITION

```
<dl>
  <dt>Ingrédients</dt>
  <dd>Farine</dd>
  <dd>Sucre</dd>
  <dt>Marche à suivre</dt>
  <dd>Mélanger...</dd>
  <dd>Verser ...</dd>
  <dt>Cuisson</dt>
  <dd>40min au four</dd>
  <dd>Laisser reposer ...</dd>
</dl>
```

```
Ingrédients
  Farine
  Sucre
Marche à suivre
  Mélanger ...
  Verser ...
Cuisson
  40min au four
  Laisser reposer ...
```

# Les liens

## LES LIENS RELATIFS

On utilise la balise `<a>` (signifie : ancre) et l'attribut "href" (référence hypertexte) le code sera donc :

```
<a href="nomdufichier.html">Lien vers le fichier</a>
```

Tout ce qui se trouve entre la balise `<a>` et `</a>` sera considéré un lien et donc, par défaut, sera souligné et en bleu.

Lorsqu'on crée un lien vers un fichier, il faut indiquer le chemin vers celui-ci.

Si on se trouve dans le même répertoire, il suffit juste d'indiquer le nom de la page html.

```
<a href="juin2008.html">Archives de juin 2008</a>
```

Si le fichier se trouve dans un autre dossier, il faut indiquer le chemin de la façon suivante :

En cas d'arborescence descendante : pour aller de la page d'accueil vers la page "juin2008.html" dans le dossier "archives" qui se trouve un niveau en-dessous :

```
<a href="archives/juin2008.html">Archives de juin 2008</a>
```

En cas d'arborescence montante : pour aller de la page archives juin 2008 vers la page d'accueil (remonter d'un niveau)

```
<a href=" ../index.html">Retour vers la page d'accueil</a>
```

"../" permet de remonter d'un niveau dans l'arborescence (de revenir dans le dossier parent).

S'il avait fallu remonter de 2 niveaux, on aurait écrit :

```
<a href=" ../../index.html">Accueil</a>
```

Pour aller d'un fichier qui se trouve dans un dossier vers un fichier qui se trouve dans un autre dossier, il faut monter pour redescendre :

```
<a href=" ../tutoriaux/photoshop.html">les tutoriaux photoshop</a>
```

Ici ../permet de sortir d'un dossier (revenir dans le dossier parent), puis il faut redescendre dans l'arborescence et indiquer dans quel dossier on veut aller : dans le dossier "tutoriaux", choisir le fichier : "photoshop.html".

## LES LIENS ABSOLUS

Le principe de création des liens absolus est le même.

Il faut ici indiquer le chemin complet, y compris le protocole utilisé (en règle générale : http).

Exemple :

```
<a href="http://www.google.fr">Faire une recherche sur Google</a>
```

## FAIRE UN LIEN QUI S'OUVRE DANS UNE NOUVELLE FENETRE

Il suffit d'ajouter l'attribut **target="\_blank"**.

```
<a href="http://www.google.fr" target="_blank">Google dans une nouvelle fenêtre</a>
```

À l'origine, l'attribut *target* permet de cibler un cadre particulier pour l'affichage de la page. Ceci dit, les cadres tentent à disparaître, du moins sur les sites dits "côté client" (ce que l'internaute voit). Certaines administrations de forums ou de sites utilisent encore les cadres car ils restent parfois pratiques. Mais nous reviendrons sur les cadres plus tard.

## LE CHOIX DES NOMS DE FICHIERS ET DE DOSSIERS

Les noms de vos fichiers et de vos dossiers constitueront vos urls. Sachant que c'est un des premiers endroits que les moteurs vont analyser pour trouver les mots-clés, il est judicieux de choisir des noms de fichiers et de dossiers en rapport avec le contenu de vos pages. Par exemple une page sur François Villon, auteur du Moyen Age aura plus de chances d'être trouvée si l'url est : **moyen-age/villon.html**, que page/page1.html Dans le premier cas, j'ai deux mots clés dans l'url. Par ailleurs, les accents, les espaces et les caractères de ponctuation sont **INTERDITS** dans les noms de dossiers et de fichiers. Il est possible d'inclure des tirets et des tirets bas. Il est recommandé de n'utiliser que des minuscules, car une fois vos pages sur un serveur, les urls deviennent sensibles à la casse. "Index.html" **N'EST PAS** la même chose que "index.html".

**FAIRE UN LIEN VERS UN EMAIL (LIEN MAILTO)**

Il s'agit d'un lien de messagerie qui fait apparaître le client de messagerie par défaut de l'utilisateur (Outlook Express sur PC ou Mail sur Mac par exemple).

```
<a href="mailto:contact@guillaumehenot.com">Contact</a>
```

Il est possible de préparer/préremplir le mail à envoyer.

Pour prédéfinir l'objet du mail, il suffit d'ajouter : ?subject=l'objet de votre mail

```
<a href="mailto:contact@guillaumehenot.com?subject=Des questions en HTML">Contact</a>
```

De même, il est possible de commencer à insérer du texte dans le corps du mail à envoyer. Ceci dit, ce ne sera que pour mettre quelques mots.

Ça se fait en ajoutant : ?body=Mon texte

```
<a href="mailto:contact@guillaumehenot.com?body=Bonjour,">Contact</a>
```

Il est même possible de combiner les 2 !

Dans ce cas, les 2 propriétés doivent être séparées par "&" :

```
<a href="mailto:contact@guillaumehenot.com?subject=Des questions en HTML&body=Bonjour,">Contact</a>
```

**FAIRE UNE ANCRE**

Il s'agit de liens internes à la page.

IL faut en premier lieu positionner un repère (l'ancre) dans la page, à l'endroit où on souhaite faire pointer le lien.

Pour ce faire, il suffit de mettre cette portion de code :

```
<a name="france"></a>
```

**Remarque** : Il n'y a pas de *href* ici. On donne simplement un nom à l'ancre.

Le lien à faire pour faire pointer à cet endroit est le suivant :

```
<a href="#france">Actu France</a>
```

Le "#" est important, il signifie de rester sur la page courante (de ne pas la recharger) et de trouver l'ancre dont le nom est "france".

On peut très bien faire pointer un lien vers une ancre dans une autre page :

```
<a href="actualite.html#france">Actu France</a>
```

Un attribut intéressant des liens est l'attribut *title*. En effet, celui-ci permet l'affichage d'une info bulle qui permet d'indiquer la fonction du lien :

```
<a href="actualite.html#france" title="Voir l'actu France">Actu France</a>
```

# Les images

## LES IMAGES SUR LE WEB

Parmi tous les formats d'images disponibles, seuls 3 sont utilisables sur le web :

- Le GIF
- Le JPG
- Le PNG

Chaque format a ses avantages et ses inconvénients.

Le GIF est l'un des plus anciens. Il ne gère que 256 couleurs, mais permet de créer de petites animations simples et permet également d'avoir UNE couleur transparente (pas de dégradé vers le transparent !). On s'en servira généralement sur de petites images, contenant des aplats de couleurs.

Plus d'infos : [http://fr.wikipedia.org/wiki/Graphics\\_Interchange\\_Format](http://fr.wikipedia.org/wiki/Graphics_Interchange_Format)

Le JPG (ou jpeg) permet, lui, de mettre en ligne des images avec des millions de couleurs (qualité photo donc). Cependant, ce format ne gère ni animation, ni transparence. C'est le format le plus répandu car les images sont très allégées. Il faut toutefois faire attention à trouver un compromis entre le poids et la qualité de l'image. En effet, celle-ci étant compressée, elle perd de la qualité qui parfois se remarque nettement.

Plus d'info : <http://fr.wikipedia.org/wiki/JPEG>

Le PNG est un format assez récent (il date tout de même des recommandations W3C de 1996 !). Il allie les avantages du JPG (millions de couleurs, compression) et la transparence du GIF. En mieux même car ce n'est pas UNE seule couleur qui devient mais ce qu'on veut. C'est une couche Alpha qui gère cette transparence, permettant ainsi, par exemple, des ombres portées ou des parties d'image translucides. Cependant, le fait d'avoir une couche supplémentaire en plus des couches RVB engendre un embonpoint et le PNG est donc plus lourd que le JPG.

De plus, Internet Explorer (uniquement lui mais y compris dans sa version 7) le gère très mal, surtout justement au niveau des transparences. C'est pourquoi ce format peine énormément à s'imposer.

Plus d'infos : [http://fr.wikipedia.org/wiki/Portable\\_Network\\_Graphics](http://fr.wikipedia.org/wiki/Portable_Network_Graphics)

## LES IMAGES : AFFICHER LES IMAGES

Il faut indiquer leurs dimensions et les afficher toujours aux dimensions d'origine.

```

```

Cela permet au navigateur de les afficher plus rapidement et éventuellement de réserver l'espace dans la page, évitant ainsi que la page ne se déforme au fur et à mesure du chargement des images (ce qui est très désagréable pour l'internaute).

### L' ATTRIBUT : ALT (TEXTE ALTERNATIF)

Alt est utilisé comme texte de légende qui remplace la photo pour les moteurs de recherches qui ne lisent pas les images et les synthétiseurs vocaux pour les malvoyants ainsi que lorsque les images ne s'affichent pas sur le navigateur.

Il apparaît également sur certains navigateurs, dans l'info bulle, lorsque le curseur est placé sur l'image.

En plus de toujours indiquer les dimensions de l'image, il faut toujours l'afficher aux dimensions d'origine. En effet, l'image étant un fichier externe à la page qui est appelée, c'est un fichier entier qui sera téléchargé. Si l'image d'origine (le fichier donc) fait 4000px sur 3000px par exemple (ex : photo d'un appareil photo) et pèse 3Mo, ce sera un fichier de 3Mo qui sera appelé et affiché sur la page web, même si l'affichage final sur la page est de 40px sur 30px.

C'est pourquoi il est important de toujours traiter ses images en amont avec un logiciel type Photoshop afin de réduire les tailles et les poids.

## LES TYPES D'IMAGE

Il y a 3 types d'image

- les images qui apportent de l'information
  - les images qui ne sont que décoratives
  - les images liens
- Pour les images qui apportent de l'information, le contenu du "alt" doit donner l'information apportée par l'image.
  - Pour les images décoratives le "alt" doit être présent, mais vide : `alt=""` (on préférera d'ailleurs placer ces images de décoration principalement dans la feuille de style).
  - Pour les images liens, le contenu doit donner la fonction du lien : `alt="aller à la section archives"`, par exemple.

L'attribut "title" permet de faire afficher des infos bulles sur les autres navigateurs.

```

```

Ici on a ajouté l'attribut "title" pour montrer l'équivalent l'info bulle.

Si les contenus de alt et de title sont différents, c'est le contenu du "title" qui sera affiché. Cependant le "alt" doit être présent !

## LES IMAGES-MAP

Il s'agit de définir des zones sensibles (cliquables) dans une image. Il faut alors calculer les coordonnées de ces zones. Il est possible de faire des zones carrées ou rectangulaires, des cercles et des polygones.

### Ici un carré, un cercle et un polygone surlignés en vert

Les coordonnées du carré/rectangle sont les coordonnées X et Y des coins supérieur gauche et inférieur droit, ici :

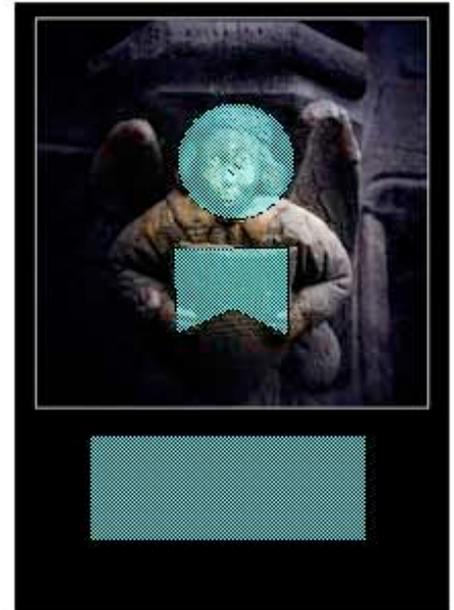
36,212,172,265 (36,212 et 172,265 donc).

Les coordonnées du cercle sont celles du centre et du rayon, ici : 108,78,29 (108,78 pour le centre et 29px de rayon).

Les coordonnées du polygone sont les coordonnées X et Y de chaque point :

80,121,133,120,133,163,106,150,79,162,77,119

L'origine 0,0 étant toujours l'angle en haut à gauche.



Code de la carte qui peut se placer n'importe où dans la corps de la page :

```
<map name="statue" id="statue">
  <area shape="circle" coords="108,78,29" href="#" />
  <area shape="poly" coords="80,121,133,120,133,163,106,150,79,162,77,119" href="#" />
  <area shape="rect" coords="36,212,172,265" href="#" />
</map>
```

Puis lorsque vous insérez l'image, *usemap="#statue"* signifie : "utilise la carte qui s'appelle #statue". Il est recommandé de rajouter l'attribut "alt" sur chacune des zones.

```

```

Les zones sont évidemment invisibles pour l'internaute. Il n'en aura connaissance qu'au passage de la souris.

# Les tableaux

Il s'agit de tableaux comme dans Word ou Excel, avec des cellules.

Ils étaient utilisés pour faire de la mise en page, avant l'apparition des boîtes et des css. Bien qu'aujourd'hui ils ne doivent plus servir pour, ils sont toujours indispensables pour insérer des données.

C'est la raison pour laquelle ils ont été créés par le W3C, il s'agit de tableau de données (table =tableau, tr= table row, ou rangée de tableau et td pour la cellule qui signifie table data ou donnée tabulaire).

La taille du tableau se détermine en pixels ou en pourcentage.

La taille fixe a pour avantage de garder la mise en page, la taille en pourcentage s'adaptera à la taille de l'écran, mais risque de modifier la mise en page. Par défaut, la taille des cellules s'adapte à leur contenu.

```
<table width="100" border="1">
  <tr>
    <td>cell1</td>
    <td>cell2</td>
  </tr>
  <tr>
    <td>cell3</td>
    <td>cell4</td>
  </tr>
</table>
```

cell1	cell2
cell3	cell4

**ATTRIBUT : ESPACEMENT DE CELLULES (CELLSPACING)**

La valeur par défaut de l'attribut d'espacement de cellules (cellspacing) est de 2 pixels :

```
<table width="100" border="0">
  <tr>
    <td>cell1</td>
    <td>cell2</td>
  </tr>
  <tr>
    <td>cell3</td>
    <td>cell4</td>
  </tr>
</table>
```

cell1	cell2
cell3	cell4

Si l'espacement des cellules est de "0", elles collent les unes aux autres.

```
<table width="100" border="0" cellspacing="0">
  <tr>
    <td>cell1</td>
    <td>cell2</td>
  </tr>
  <tr>
    <td>cell3</td>
    <td>cell4</td>
  </tr>
</table>
```

cell1	cell2
cell3	cell4

Si l'espacement des cellules est de "10", elles seront séparées par 10 pixels de tous les côtés.

```
<table width="100" border="0" cellspacing="10">
  <tr>
    <td>cell1</td>
    <td>cell2</td>
  </tr>
  <tr>
    <td>cell3</td>
    <td>cell4</td>
  </tr>
</table>
```

cell1	cell2
cell3	cell4

**ATTRIBUT : LE REMPLISSAGE DANS LES CELLULES (CELLPADDING)**

La valeur par défaut de l'attribut de remplissage de cellules est de 1 pixel :

```
<table width="100" border="0">
  <tr>
    <td>cell1</td>
    <td>cell2</td>
  </tr>
  <tr>
    <td>cell3</td>
    <td>cell4</td>
  </tr>
</table>
```

cell1	cell2
cell3	cell4

Si le remplissage dans les cellules est de "0", le contenu colle aux bords.

```
<table width="100" border="0" cellpadding="0">
  <tr>
    <td>cell1</td>
    <td>cell2</td>
  </tr>
  <tr>
    <td>cell3</td>
    <td>cell4</td>
  </tr>
</table>
```

cell1	cell2
cell3	cell4

Si l'espacement des cellules est de "10", il y aura 10 pixels autour du texte.

```
<table width="100" border="0" cellpadding="10">
  <tr>
    <td>cell1</td>
    <td>cell2</td>
  </tr>
  <tr>
    <td>cell3</td>
    <td>cell4</td>
  </tr>
</table>
```

cell1	cell2
cell3	cell4

Il est bien sûr possible de cumuler les 2.

```
<table width="100" border="0" cellpadding="10" cellspacing="10">
  <tr>
    <td>cell1</td>
    <td>cell2</td>
  </tr>
  <tr>
    <td>cell3</td>
    <td>cell4</td>
  </tr>
</table>
```

cell1	cell2
cell3	cell4

## Les formulaires

Nous traitons ici la structure du formulaire en html, avec un envoi des données se faisant par email.

Les attributs principaux de la balise <form> sont "action" et "method". Ils sont indispensables. D'autres attributs viennent éventuellement s'ajouter ensuite.

**method="post"** : les données n'apparaissent pas dans l'url, elles sont transmises de façon masquée. C'est le plus discret / sécurisé.

**method="get"** : passe les données via l'url, elles sont donc visibles par l'internaute.

**action="mailto:votreadresseemail"** : (si vous envoyez votre formulaire par email)

**enctype="text/plain"** : permet de signaler le type de texte envoyé (ici texte brut). On peut aussi trouver "text/html".

Ci-dessous le code complet :

```
<form method="post" action="mailto:contact@guillaumehenot.com?subject=Mes questions en html
enctype="text/plain">
    .....
</form>
```

## Éléments de formulaire

### LE CHAMP TEXTE NORMAL

C'est le champ le plus classique, celui qui permet d'entrer du texte comme un nom ou un prénom par exemple

```
<input type="text" name="prenom" value="" />
```

### LES BOUTONS RADIO

Ils ne permettent qu'un seul choix parmi la série du même nom. L'affichage du bouton radio diffère en fonction des navigateurs.

```
<input type="radio" name="abonnement" value="oui" /> oui
<input type="radio" name="abonnement" value="non" /> non
```

### LES CASES A COCHER

Elles permettent un ou plusieurs choix dans une série. De même que les bouton radio, leur affichage diffère selon les navigateurs.

```
<input type="checkbox" name="interet1" value="musique" /> Musique
<input type="checkbox" name="interet2" value="danse" /> Danse
<input type="checkbox" name="interet3" value="informatique" /> Informatique
```

## LA LISTE DEROULANTE

Elle permet un choix dans un menu déroulant. L'affichage de ce menu diffère lui aussi selon les navigateurs, mais il est toutefois en partie "stylable" avec des CSS.

```
<select name="pays">
  <option value="France">France</option>
  <option value="Allemagne">Allemagne</option>
  <option value="Espagne">Espagne</option>
  <option value="Italie">Italie</option>
</select>
```

Par défaut, c'est le premier élément de la liste qui apparaît sur la page. On peut cependant en pré-sélectionner un en particulier ("espagne" ici) :

```
<select name="pays">
  <option value="France">france</option>
  <option value="Allemagne">Allemagne</option>
  <option value="Espagne" selected="selected">Espagne</option>
  <option value="Italie">italie</option>
</select>
```

## LA BOITE DE DIALOGUE

Elle permet d'écrire des commentaires sans limitation de nombre de caractères, l'ascenseur se déroule en bout de boîte.

```
<textarea name="commentaires" cols="20" rows="4" id="commentaires"></textarea>
```

## LES BOUTONS

Pour envoyer le formulaire et annuler les données inscrites dans le formulaire.

```
<input type="submit" name="Submit" value="Envoyer" />
<input type="reset" name="Reset" value="Annuler" />
```

## LES CHAMPS CACHES

Il est possible d'insérer des champs invisibles pour l'utilisateur dont la valeur sera bien envoyée avec le reste du formulaire.

```
<input type="hidden" name="invisible" value="La valeur cachée" />
```

## LE CHAMP PASSWORD

Il est identique au champ texte, mais masque son contenu par des ●●●●●●●●

```
<input type="password" name="pass" value="" />
```

## LES LABELS

Les labels sont des "étiquettes" attachées aux champs du formulaire.

L'une des utilités directes est de faciliter la saisie des données car un clic sur le texte contenu dans la balise `<label>` permet d'activer le champ. C'est donc particulièrement intéressant pour les boutons radio ou les cases à cocher qui sont assez petites et parfois difficiles à cliquer pour certaines personnes.

En outre, la balise `<label>` est un élément qu'il sera facile de styler afin d'améliorer la mise en page du formulaire.

La balise `<label>` peut soit entourer le champ, soit être placée ailleurs (dans une autre case d'un tableau par exemple) et faire référence à lui.

```
<label>Prénom : <input type="text" name="prenom" /></label>
<table>
  <tr>
    <td><label for="prenom">Prénom</label></td>
    <td><input type="text" name="prenom" id="prenom" /></td>
  </tr>
</table>
```

Si la balise `<label>` est placée ailleurs, il faudra lui indiquer dans son attribut "for" le ID du champ auquel elle fait référence (en rouge ci-dessus)

## Les cadres

Auparavant, les navigateurs ne pouvaient afficher qu'un seul fichier HTML mais avec la technologie développée par Netscape, on peut désormais diviser en plusieurs zones la page affichée par le navigateur : ces zones ou fenêtres ou encore cadres se nomment frames. Les frames sont apparus avec la version 3.0. Même si bon nombre de Webmasters les utilisaient, ils n'étaient pas une composante officielle du langage.

S'ils ne sont plus trop utilisés sur les sites "côté client", on les trouve encore parfois sur certaines applications (chat, etc ...), ou administration de sites ou de forums.

Une page de cadres ne contient également que 2 parties : le <head> et le <frameset> (il n'y a plus de <body> !!). Il faudra de plus utiliser le bon Doctype, par exemple :

```
<!DOCTYPEhtml PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

### LE CONTENEUR <FRAMESET>

La balise <frameset> prend la place du tag <body>, c'est elle qui dans un premier temps va définir les cadres, qu'ils soient verticaux ou horizontaux, et leurs dimensions (en % ou en pixels).

Voici deux exemples (les plus courants) ainsi que le code source.

- Le premier contient deux cadres horizontaux appelés Zone 1 et Zone2
- Le deuxième contient aussi deux cadres horizontaux appelés Zone 1 et Zone2 et un cadre vertical appelé Zone 3

Exemple 1 :

```
<frameset cols="20%,80%">
  <frame src="zone1.htm" name="zone1" />
  <frame src="zone2.htm" name="zone2" />
</frameset>
```

Exemple 2 :

```
<frameset rows="80%,20%">
  <frameset cols="20%,80%">
    <frame src="zone1.htm" name="zone1" />
    <frame src="zone2.htm" name="zone2" />
  </frameset>
  <frame src="zone3.htm" name="zone3" />
</frameset>
```

## Attributs du conteneur <frameset>

### LES ATTRIBUTS ROWS ET COLS :

Les attributs rows et cols définissent le type de cadre, qui peut être vertical (rows - rangées) ou horizontal (cols - colonnes). Ils prennent une liste de valeurs séparées par des virgules. Ces valeurs sont exprimées en pourcentage entre 1 et 100 (valeurs relatives) ou en pixels (valeurs absolues).

Deux remarques : si une valeur manque ou est remplacée par une étoile (\*) la colonne ou rangée s'ajustera automatiquement. De plus, la hauteur de toutes les rangées ou la largeur de toutes les colonnes doit correspondre à la hauteur ou à la largeur de la fenêtre du navigateur, sinon la dernière valeur ne sera pas prise en compte et la fenêtre ou rangée s'ajustera automatiquement à l'écran.

Exemple de deux colonnes, qui ont comme valeurs 20% et 80 %.

```
<frameset cols="20%,80%">
```

Exemple de deux colonnes, l'une d'une largeur de 50 pixels et l'autre s'ajustant automatiquement à la fenêtre du navigateur.

```
<frameset cols="50,*">
```

### L'ATTRIBUT FRAMEBORDER

Cet attribut permet de déterminer si les cadres auront ou n'auront pas de bordures. Il a deux valeurs yes ou no.

frameborder="yes" ou frameborder="no"

```
<frameset cols="20%,80%" frameborder="no">
```

### L'ATTRIBUT BORDER

Cet attribut permet de déterminer la taille des bordures entourant les cadres. Il peut prendre plusieurs valeurs ex: border=n, n étant une valeur en pixels définissant la taille de la bordure. La valeur 0 indique "aucune bordure".

border="0" ou par exemple border="5"

```
<frameset cols="20%,80%" border="5">
```

**L'ATTRIBUT BORDERCOLOR**

Cet attribut permet de déterminer la couleur de l'ensemble des bordures des cadres. Il suffit pour cela de spécifier une couleur sous forme de nom ou de sa valeur hexadécimale .

bordercolor="red" ou bordercolor="#ff0000"

```
<frameset cols="20%,80%" bordercolor="red">
```

**L'ATTRIBUT FRAMESPACING**

Cet attribut permet de déterminer un espace entre les cadres. Il peut prendre plusieurs valeurs ex: framespacing=n, n étant une valeur en pixels définissant l'espace entre les cadres. La valeur 0 indique "aucun espace".

framespacing="0" ou par exemple framespacing="5"

```
<frameset cols="20%,80%" framespacing="5">
```

**LA BALISE <FRAME>**

La commande frame permet de définir un cadre à l'intérieur du conteneur <frameset>. Elle a plusieurs attributs, les plus importants étant src et name.

**L'ATTRIBUT SRC**

Cet attribut indique l'URL (chemin) du document HTML qui sera affiché dans un cadre spécifique.

```
<frame src="zone1.htm" />
```

**L'ATTRIBUT NAME**

Cet attribut permet de donner un nom à un cadre, ce qui permettra ensuite de l'appeler avec l'attribut target (attention, majuscules et minuscules sont prises en compte !).

```
<frame src="zone1.htm" name="zone1" />
```

**LES ATTRIBUTS MARGINWIDTH ET MARGINHEIGHT**

- marginwidth permet de spécifier la grandeur des marges de gauche et de droite du cadre créé, la valeur doit être exprimée en pixels, elle peut avoir comme valeur 0.

- marginheight permet de spécifier la grandeur des marges de haut et de bas du cadre créé, la valeur doit être exprimée en pixels, elle peut avoir comme valeur 0.

```
<frame src="zone1.htm" name="zone1" marginwidth="2" marginheight="2" />
```

**L'ATTRIBUT FRAMEBORDER**

Cet attribut permet de déterminer si les cadres auront ou n'auront pas de bordure. Deux valeurs yes ou no.

```
<frame src="zone1.htm" name="zone1" frameborder="no" />
```

**L'ATTRIBUT BORDER (NETSCAPE UNIQUEMENT)**

Cet attribut permet d'ajuster l'espace entre les cadres. La valeur doit être exprimée en pixels.

```
<frame src="zone1.htm" name="zone1" border="2" />
```

**L'ATTRIBUT NORESIZE**

En utilisant cet attribut, vous interdisez à l'utilisateur de redimensionner les cadres. Par défaut les cadres peuvent être redimensionnés.

```
<frame src="zone1.htm" name="zone1" noresize />
```

**L'ATTRIBUT SCROLLING**

Cet attribut permet d'attribuer ou non une barre de défilement (scrollbar) à un cadre. Il possède trois valeurs :

- yes : Indique que la barre de défilement sera toujours visible.
- no : Indique que la barre de défilement ne sera jamais visible (à tester avant de l'utiliser).
- auto : Indique que le navigateur déterminera si la barre de défilement est nécessaire.

```
<frame src="zone1.htm" name="zone1" scrolling="auto" />
```

## LES NAVIGATEURS NON COMPATIBLES

Même s'il est de plus en plus rare de trouver des navigateurs non compatibles, il faut néanmoins ne pas négliger cet aspect, sous peine de se passer de quelques visiteurs. La technique consiste à utiliser les balises `<noframes>` et `</noframes>`. Ces balises permettent de spécifier un texte HTML en version normale. Entre ces balises, il faut donc, théoriquement, développer un deuxième site : vous pouvez pour simplifier le travail préciser des regrets polis et insérer des liens vers les sites officiels de navigateurs récents pour que le visiteur opte pour une mise à jour.

Exemple d'utilisation des balises `<noframes>` et `</noframes>` :

```
<frameset rows="20%,80%">
  <frame src="zone1.htm" name="zone1" />
  <frame src="zone2.htm" name="zone2" />
</frameset>
<noframes>
<body>
  Désolé mais ce site comme vous pouvez le voir utilise la technique des frames. Merci de faire la
  mise à jour de votre navigateur.
</body>
</noframes>
```

## Cadres locaux <iframe>

La technique des cadres locaux s'obtient avec la balise <iframe> ; elle permet d'insérer une fenêtre à n'importe quel endroit de votre document. Cette fenêtre fera référence à un autre document HTML. La balise <iframe> doit être dans le corps de votre document, c'est-à-dire entre les balises <body> et </body>. Attention : cette balise est spécifique à Internet Explorer malgré son approbation par le W3C et sa prise en compte par les autres navigateurs. Elle a les mêmes attributs que la balise <frame> c'est-à-dire :

- border
- bordercolor
- frameborder
- marginheight
- marginwidth
- name
- noresize
- scrolling
- src

Elle possède en plus les attributs height et width qui déterminent la hauteur et la largeur du cadre inséré dans votre page. Valeur en pixels.

Note : Un texte de remplacement (pour les navigateurs qui ne la prendraient pas en compte) peut être mis entre <iframe> et </iframe>.

Exemple d'un cadre local.

```
<iframe src="mon_iframe.htm" name="zone1" height="80" width="150">Cadre local</iframe>
```

```
<iframe src="../divers/mon_iframe.php" name="zone1" height="80" width="150">Cadre local</iframe>
```

## LIENS VERS D'AUTRES FRAMES

Normalement, les pages appelées par des liens s'affichent dans la frame où les liens ont été activés. L'attribut **target** permet d'appeler un autre cadre par le nom défini avec l'attribut name précédemment. Mais il existe des noms réservés de frame cible dont voici les différentes valeurs.

Valeur	Signification
_self	L'URL cible va s'afficher dans la même frame que le lien
_parent	L'URL cible va s'afficher dans la structure externe
_blank	L'URL cible va s'afficher dans une nouvelle fenêtre ouverte par le navigateur
_top	L'URL cible va s'afficher dans toute la fenêtre du navigateur et les frames disparaissent
_new	Identique à _blank (nouvelle fenêtre)

On peut bien sûr entrer dans l'attribut *target* le nom du cadre dans lequel on souhaite ouvrir le lien :

```
<a href="page.htm" target="zone2">Lien</a>
```

Exemple de lien externe avec la valeur \_top.

```
<a href="http://www.allhtml.com" target="_top">Lien</a>
```

## ANNEXE 1 : Caractères spéciaux HTML

La prise en charge des caractères suivants est effective à partir de IE 7, FireFox 1 et Netscape 4. Si le caractère est compatible avec une version antérieure, c'est indiqué dans la colonne "compatibilité". Liste non-exhaustive et sous réserve de compatibilité avec certains navigateurs (il faut tester !).

### CARACTERES GENERAUX

caractère	code texte	code numérique	commentaire	compatibilité
	&nbsp;	&#160;	espace insécable	NS1
-	&shy;	&#173;	tiret de césure optionnelle(permets au navigateur de couper le mot au bon endroit si besoin de passer à la ligne)	
⌈	&lrm;	&#8206;	marque gauche-à-droite	pas Win98
⌋	&rlm;	&#8207;	marque droite-à-gauche	pas Win98
"	&quot;	&#34;	guillemet anglais, guillemet droit (quote)	XML, NS1
«	&laquo;	&#171;	guillemet français ouvrant	NS3
»	&raquo;	&#187;	guillemet français fermant	NS3
‹	&lsaquo;	&#8249;	guillemet français simple ouvrant	
›	&rsaquo;	&#8250;	guillemet français simple fermant	
“	&ldquo;	&#8220;	guillemet double ouvrant, guillemet-apostrophe double culbuté	
”	&rdquo;	&#8221;	guillemet double fermant, guillemet-apostrophe double	
„	&bdquo;	&#8222;	guillemet double fermant bas, guillemet-virgule double inférieur	
'	&apos;	&#39;	guillemet simple droit	XML
‘	&lsquo;	&#8216;	guillemet simple ouvrant, guillemet-apostrophe culbuté	
’	&rsquo;	&#8217;	guillemet simple fermant, guillemet-apostrophe	
‚	&sbquo;	&#8218;	guillemet simple fermant bas, guillemet-virgule inférieur	
...	&hellip;	&#8230;	points de suspension	
!	!	&#33;	point d'exclamation	
¡	&iexcl;	&#161;	point d'exclamation inversé	NS3
?	?	&#63;	point d'interrogation	
¿	&iquest;	&#191;	point d'interrogation inversé	NS3
(	(	&#40;	parenthèse ouvrante	
)	)	&#41;	parenthèse fermante	
[	[	&#91;	crochet ouvrant	
]	]	&#93;	crochet fermant	
{	{	&#123;	accolade ouvrante	
}	}	&#125;	accolade fermante	
¨	&uml;	&#168;	tréma	NS3
´	&acute;	&#180;	accent aigu	NS3
`	`	&#96;	accent grave	
^	^	&#94;	accent circonflexe	
ˆ	&circ;	&#710;	accent circonflexe	
˜	˜	&#126;	tilde	
˘	&tilde;	&#732;	petit tilde	
¸	&cedil;	&#184;	cédille	NS3
#	#	&#35;	dièse	

*	*	&#42;	étoile	
,	,	&#44;	virgule	
.	.	&#46;	point	
:	:	&#58;	deux-points	
;	;	&#59;	point-virgule	
.	&#middot;	&#183;	point médian	NS3, pas Win98
•	&#bull;	&#8226;	gros point médian	
–	&#macr;	&#175;	macron	NS3
–	&#oline;	&#8254;	tiret en chef (overline, spacing overscore)	pas IE5, pas Amaya8, pas Win98
-	-	&#45;	tiret, tiret quart-cadratin, tiret de mots composés	
–	&#ndash;	&#8211;	tiret demi-cadratin, tiret d'incise	
—	&#mdash;	&#8212;	tiret cadratin, tiret de dialogue	
–	–	&#95;	tiret bas (underscore)	
		&#124;	séparateur vertical (pipe)	
‡	&#brvbar;	&#166;	barre verticale scindée	NS3
	&#zwnj;	&#8204;	antiliant sans chasse	pas Win98
	&#zwj;	&#8205;	liant sans chasse	pas Win98
†	&#dagger;	&#8224;	obèle	pas Amaya8
‡	&#Dagger;	&#8225;	double obèle	pas Amaya8
§	&#sect;	&#167;	section	NS3
¶	&#para;	&#182;	paragraphe	NS3
©	&#copy;	&#169;	copyright	NS1
®	&#reg;	&#174;	marque déposée	NS1
™	&#trade;	&#8482;	marque commerciale (trade mark)	
&	&#amp;	&#38;	et commercial, esperluette, ampersand	XML, NS1
@	@	&#64;	arobase (chez)	
/	/	&#47;	diviser (slash)	
\	\	&#92;	anti-slash	
◊	&#loz;	&#9674;	losange	pas IE5
♠	&#spades;	&#9824;	pique noir	pas IE5
♣	&#clubs;	&#9827;	trèfle noir	pas IE5
♥	&#hearts;	&#9829;	cœur noir	pas IE5
♦	&#diams;	&#9830;	carreau noir	pas IE5
←	&#larr;	&#8592;	flèche vers la gauche	pas IE5
↑	&#uarr;	&#8593;	flèche vers le haut	pas IE5
→	&#rarr;	&#8594;	flèche vers la droite	pas IE5
↓	&#darr;	&#8595;	flèche vers le bas	pas IE5
↔	&#harr;	&#8596;	flèche bilatérale gauche-droite	pas IE5

**MONNAIES**

caractère	code texte	code numérique	commentaire	compatibilité
¤	&curren;	&#164;	symbole monétaire	NS3
€	&euro;	&#128;	euro (monnaie européenne)	NS4
\$	\$	&#36;	dollar (monnaie américaine)	
¢	&cent;	&#162;	cent (monnaie américaine)	NS3
£	&pound;	&#163;	pound (Livre sterling, monnaie anglaise)	NS3
¥	&yen;	&#165;	yen (monnaie japonaise)	NS3
f	&fnof;	&#402;	florin (idem fonction)	

**CARACTERES ALPHABETIQUES ACCENTUES ET SPECIAUX**

caractère	code texte	code numérique	commentaire	compatibilité
á	&aacute;	&#225;		NS1
Á	&Aacute;	&#193;		NS1
â	&acirc;	&#226;		NS1
Â	&Acirc;	&#194;		NS1
à	&agrave;	&#224;		NS1
À	&Agrave;	&#192;		NS1
å	&aring;	&#229;		NS1
Å	&Aring;	&#197;		NS1
ã	&atilde;	&#227;		NS1
Ã	&Atilde;	&#195;		NS1
ä	&auml;	&#228;		NS1
Ä	&Auml;	&#196;		NS1
æ	&aelig;	&#230;		NS1
Æ	&AElig;	&#198;		NS1
ç	&ccedil;	&#231;		NS1
Ç	&Ccedil;	&#199;		NS1
é	&eacute;	&#233;		NS1
É	&Eacute;	&#201;		NS1
ê	&ecirc;	&#234;		NS1
Ê	&Ecirc;	&#202;		NS1
è	&egrave;	&#232;		NS1
È	&Egrave;	&#200;		NS1
ë	&euml;	&#235;		NS1
Ë	&Euml;	&#203;		NS1
í	&iacute;	&#237;		NS1
Í	&Iacute;	&#205;		NS1
î	&icirc;	&#238;		NS1
Î	&Icirc;	&#206;		NS1
ì	&igrave;	&#236;		NS1
Ì	&Igrave;	&#204;		NS1
ï	&iuml;	&#239;		NS1
Ï	&Iuml;	&#207;		NS1
ñ	&ntilde;	&#241;		NS1
Ñ	&Ntilde;	&#209;		NS1
ó	&oacute;	&#243;		NS1
Ó	&Oacute;	&#211;		NS1
ô	&ocirc;	&#244;		NS1
Ô	&Ocirc;	&#212;		NS1
ò	&ograve;	&#242;		NS1
Ò	&Ograve;	&#210;		NS1

ø	&oslash;	&#248;		NS1
Ø	&Oslash;	&#216;		NS1
ō	&otilde;	&#245;		NS1
Õ	&Otilde;	&#213;		NS1
ö	&ouml;	&#246;		NS1
Ö	&Ouml;	&#214;		NS1
œ	&oelig;	&#339;	ligature minuscule latine oe	pas NS4
Œ	&OElig;	&#338;	ligature majuscule latine OE	pas NS4
š	&scaron;	&#353;	lettre minuscule latine s avec caron	pas NS4
Š	&Scaron;	&#352;	lettre majuscule latine S avec caron	pas NS4
ß	&szlig;	&#223;	lettre minuscule allemande s dur	NS1
ð	&eth;	&#240;	lettre minuscule islandaise ed	NS1
Ð	&ETH;	&#208;	lettre majuscule islandaise ED	NS1
þ	&thorn;	&#254;	lettre minuscule islandaise thorn	NS1
Þ	&THORN;	&#222;	lettre majuscule islandaise Thorn	NS1
ú	&uacute;	&#250;		NS1
Ú	&Uacute;	&#218;		NS1
û	&ucirc;	&#251;		NS1
Û	&Ucirc;	&#219;		NS1
ù	&ugrave;	&#249;		NS1
Û	&Ugrave;	&#217;		NS1
ü	&uuml;	&#252;		NS1
Ü	&Uuml;	&#220;		NS1
ý	&yacute;	&#253;		NS1
Ý	&Yacute;	&#221;		NS1
ÿ	&yuml;	&#255;		NS1
ÿ	&Yuml;	&#376;		pas NS4

## ALPHABET GREC

caractère	code texte	code numérique	commentaire	compatibilité
α	&alpha;	&#945;	alpha	
A	&Alpha;	&#913;	Alpha	
β	&beta;	&#946;	beta	
B	&Beta;	&#914;	Beta	
γ	&gamma;	&#947;	gamma	
Γ	&Gamma;	&#915;	Gamma	
δ	&delta;	&#948;	delta	
Δ	&Delta;	&#916;	Delta	
ε	&epsilon;	&#949;	epsilon	
E	&Epsilon;	&#917;	Epsilon	
ζ	&zeta;	&#950;	zeta	
Z	&Zeta;	&#918;	Zeta	
η	&eta;	&#951;	eta	
H	&Eta;	&#919;	Eta	
θ	&theta;	&#952;	theta	
Θ	&Theta;	&#920;	Theta	
ι	&iota;	&#953;	iota	
I	&Iota;	&#921;	Iota	
κ	&kappa;	&#954;	kappa	
K	&Kappa;	&#922;	Kappa	
λ	&lambda;	&#955;	lambda	
Λ	&Lambda;	&#923;	Lambda	
μ	&mu;	&#956;	mu	
M	&Mu;	&#924;	Mu	
ν	&nu;	&#957;	nu	
N	&Nu;	&#925;	Nu	
ξ	&xi;	&#958;	xi	
Ξ	&Xi;	&#926;	Xi	
ο	&omicron;	&#959;	omicron	
O	&Omicron;	&#927;	Omicron	
π	&pi;	&#960;	pi	
Π	&Pi;	&#928;	Pi	
ρ	&rho;	&#961;	rho	
P	&Rho;	&#929;	Rho	
σ	&sigma;	&#963;	sigma	
ς	&sigmaf;	&#962;	sigma final	
Σ	&Sigma;	&#931;	Sigma	
τ	&tau;	&#964;	tau	
T	&Tau;	&#932;	Tau	
υ	&upsilon;	&#965;	upsilon	
Υ	&Upsilon;	&#933;	Upsilon	
φ	&phi;	&#966;	phi	
Φ	&Phi;	&#934;	Phi	
χ	&chi;	&#967;	chi	
X	&Chi;	&#935;	Chi	
ψ	&psi;	&#968;	psi	
Ψ	&Psi;	&#936;	Psi	
ω	&omega;	&#969;	omega	
Ω	&Omega;	&#937;	Omega	

## SCIENCES

caractère	code texte	code numérique	commentaire	compatibilité
°	&deg;	&#176;	degré	NS3
μ	&micro;	&#181;	mu (lettre grecque)	NS3
<	&lt;	&#60;	inférieur (less-than)	XML, NS1
>	&gt;	&#62;	supérieur (greater-than)	XML, NS1
≤	&le;	&#8804;	inférieur ou égal (less or equal)	pas IE5
≥	&ge;	&#8805;	supérieur ou égal (greater or equal)	pas IE5
=	=	&#61;	égal	
≈	&asymp;	&#8776;	presque égal (asymptotic)	pas IE5
≠	&ne;	&#8800;	différent (not equal)	pas IE5
≡	&equiv;	&#8801;	équivalent	pas IE5
±	&plusmn;	&#177;	plus-ou-moins	NS3
-	&minus;	&#8722;	moins	pas IE5
+	+	&#43;	plus	
×	&times;	&#215;	multiplication	NS3
÷	&divide;	&#247;	division	NS3
/	&frasl;	&#8260;	fraction	pas IE5
%	%	&#37;	pour cent	
‰	&permil;	&#8240;	pour mille	pas Amaya8
¼	&frac14;	&#188;	un quart	NS3
½	&frac12;	&#188;	un demi	NS3
¾	&frac34;	&#188;	trois quarts	NS3
<sup>1</sup>	&sup1;	&#185;	exposant 1	NS3
<sup>2</sup>	&sup2;	&#178;	exposant 2	NS3
<sup>3</sup>	&sup3;	&#179;	exposant 3	NS3
º	&ordm;	&#186;	indicateur ordinal masculin	NS3
ª	&ordf;	&#170;	indicateur ordinal féminin	NS3
f	&fnof;	&#402;	fonction (idem florin)	
'	&prime;	&#8242;	prime, minutes, dérivée	pas IE5
''	&Prime;	&#8243;	double prime, secondes, dérivée seconde	pas IE5
∂	&part;	&#8706;	dérivée partielle	pas IE5
∏	&prod;	&#8719;	produit n-aire	pas IE5
∑	&sum;	&#8721;	somme n-aire	pas IE5
√	&radic;	&#8730;	racine carrée	pas IE5
∞	&infin;	&#8734;	infini	pas IE5
¬	&not;	&#172;	crochet de négation	NS3
∩	&cap;	&#8745;	intersection	pas IE5
∫	&int;	&#8747;	intégrale	pas IE5

**CARACTERES DIVERS :**

Leur utilisation n'est pas recommandée car ils sont moins bien reconnus par Internet Explorer 7 ; leur affichage y est néanmoins possible en utilisant une police unicode telle "Arial Unicode MS", par exemple. De même, pour Opera 7. Notez que Mozilla depuis la version 1, Netscape depuis la version 7, Opera depuis la version 8, les prennent tous en charge.

**RECONNUS SOUS IE6.0+ AVEC UNE POLICE UNICODE**

caractère	code texte	code numérique	commentaire	compatibilité
⇒	&rArr;	&#8658;	rightwards double arrow	IE6.0+Unicode
⇔	&hArr;	&#8660;	left right double arrow	IE6.0+Unicode
∀	&forall;	&#8704;	for all	IE6.0+Unicode
∃	&exist;	&#8707;	there exists	IE6.0+Unicode
∇	&nabla;	&#8711;	nabla = backward difference	IE6.0+Unicode
∈	&isin;	&#8712;	element of	IE6.0+Unicode
⊃	&ni;	&#8715;	contains as member	IE6.0+Unicode , pas Amaya8
∝	&prop;	&#8733;	proportional to	IE6.0+Unicode
∠	&ang;	&#8736;	angle	IE6.0+Unicode
∧	&and;	&#8743;	logical and = wedge	IE6.0+Unicode
∨	&or;	&#8744;	logical or = vee	IE6.0+Unicode
∪	&cup;	&#8746;	union = cup	IE6.0+Unicode
∴	&there4;	&#8756;	therefore	IE6.0+Unicode
~	&sim;	&#8764;	tilde operator = varies with = similar to	IE6.0+Unicode
⊂	&sub;	&#8834;	subset of	IE6.0+Unicode
⊃	&sup;	&#8835;	superset of	IE6.0+Unicode
⊆	&sube;	&#8838;	subset of or equal to	IE6.0+Unicode
⊇	&supe;	&#8839;	superset of or equal to	IE6.0+Unicode
⊥	&perp;	&#8869;	up tack = orthogonal to = perpendicular	IE6.0+Unicode

## RECONNUS SOUS IE6.1+ AVEC UNE POLICE UNICODE

caractère	code texte	code numérique	commentaire	compatibilité
⊕	&oplus;	&#8853;	circled plus = direct sum	IE6.1+Unicode
	&thinsp;	&#8201;	espace fine	IE6.1+Unicode
	&ensp;	&#8194;	espace demi-cadratin	IE6.1+Unicode
	&emsp;	&#8195;	espace cadratin	IE6.1+Unicode
ℙ	&weierp;	&#8472;	script capital P = power set = Weierstrass p	IE6.1+Unicode
ℑ	&image;	&#8465;	partie imaginaire (majuscule I gothique)	IE6.1+Unicode
ℜ	&real;	&#8476;	partie réelle (majuscule R gothique)	IE6.1+Unicode
↵	&crarr;	&#8629;	downwards arrow with corner leftwards = carriage return	IE6.1+Unicode
⇐	&lArr;	&#8656;	leftwards double arrow	IE6.1+Unicode
⇑	&uArr;	&#8657;	upwards double arrow	IE6.1+Unicode
⇓	&dArr;	&#8659;	downwards double arrow	IE6.1+Unicode
∅	&empty;	&#8709;	empty set = null set = diameter	IE6.1+Unicode
∉	&notin;	&#8713;	not an element of	IE6.1+Unicode
*	&lowast;	&#8727;	asterisk operator	IE6.1+Unicode
≈	&cong;	&#8773;	approximately equal to	IE6.1+Unicode
⊄	&nsup;	&#8836;	not a subset of	IE6.1+Unicode
⊗	&otimes;	&#8855;	circled times = produit vectoriel	IE6.1+Unicode
·	&sdot;	&#8901;	dot operator	IE6.1+Unicode
⌈	&lceil;	&#8968;	left ceiling = apl upstile	IE6.1+Unicode
⌋	&rceil;	&#8969;	right ceiling	IE6.1+Unicode
⌊	&lfloor;	&#8970;	left floor = apl downstile	IE6.1+Unicode
⌋	&rfloor;	&#8971;	right floor	IE6.1+Unicode
⟨	&lang;	&#9001;	chevron vers la gauche (left-pointing angle bracket) = bra	IE6.1+Unicode
⟩	&rang;	&#9002;	chevron vers la droite (right-pointing angle bracket) = ket	IE6.1+Unicode
ℵ	&alefsym;	&#8501;	alef symbol = first transfinite cardinal	IE6.1+Unicode , pas Opera7
ϑ	&thetasym;	&#977;	symbole theta	
ϖ	&piv;	&#982;	symbole pi	
Υ	&upsih;	&#978;	symbole Upsilon crochet	

# Annexe 2 : les couleurs HTML "Web Safe"

	#000000		#330000		#660000		#990000		#CC0000		#FF0000
	#003300		#333300		#663300		#993300		#CC3300		#FF3300
	#006600		#336600		#666600		#996600		#CC6600		#FF6600
	#009900		#339900		#669900		#999900		#CC9900		#FF9900
	#00CC00		#33CC00		#66CC00		#99CC00		#CCC000		#FFCC00
	#00FF00		#33FF00		#66FF00		#99FF00		#CCFF00		#FFFF00
	#000033		#330033		#660033		#990033		#CC0033		#FF0033
	#003333		#333333		#663333		#993333		#CC3333		#FF3333
	#006633		#336633		#666633		#996633		#CC6633		#FF6633
	#009933		#339933		#669933		#999933		#CC9933		#FF9933
	#00CC33		#33CC33		#66CC33		#99CC33		#CCC033		#FFCC33
	#00FF33		#33FF33		#66FF33		#99FF33		#CCFF33		#FFFF33
	#000066		#330066		#660066		#990066		#CC0066		#FF0066
	#003366		#333366		#663366		#993366		#CC3366		#FF3366
	#006666		#336666		#666666		#996666		#CC6666		#FF6666
	#009966		#339966		#669966		#999966		#CC9966		#FF9966
	#00CC66		#33CC66		#66CC66		#99CC66		#CCC066		#FFCC66
	#00FF66		#33FF66		#66FF66		#99FF66		#CCFF66		#FFFF66
	#000099		#330099		#660099		#990099		#CC0099		#FF0099
	#003399		#333399		#663399		#993399		#CC3399		#FF3399
	#006699		#336699		#666699		#996699		#CC6699		#FF6699
	#009999		#339999		#669999		#999999		#CC9999		#FF9999
	#00CC99		#33CC99		#66CC99		#99CC99		#CCC099		#FFCC99
	#00FF99		#33FF99		#66FF99		#99FF99		#CCFF99		#FFFF99
	#0000CC		#3300CC		#6600CC		#9900CC		#CC00CC		#FF00CC
	#0033CC		#3333CC		#6633CC		#9933CC		#CC33CC		#FF33CC
	#0066CC		#3366CC		#6666CC		#9966CC		#CC66CC		#FF66CC
	#0099CC		#3399CC		#6699CC		#9999CC		#CC99CC		#FF99CC
	#00CCCC		#33CCCC		#66CCCC		#99CCCC		#CCC0CC		#FFCCCC
	#00FFCC		#33FFCC		#66FFCC		#99FFCC		#CCFFCC		#FFFFCC
	#0000FF		#3300FF		#6600FF		#9900FF		#CC00FF		#FF00FF
	#0033FF		#3333FF		#6633FF		#9933FF		#CC33FF		#FF33FF
	#0066FF		#3366FF		#6666FF		#9966FF		#CC66FF		#FF66FF
	#0099FF		#3399FF		#6699FF		#9999FF		#CC99FF		#FF99FF
	#00CCFF		#33CCFF		#66CCFF		#99CCFF		#CCC0FF		#FFCCFF
	#00FFFF		#33FFFF		#66FFFF		#99FFFF		#CCFFFF		#FFFFFF

Web-safe Colours

	Black		Maroon		Green		Navy	FF - 255	77 - 119
#000000		#800000		#008000		#000080		EE - 238	66 - 102
	Silver		Red		Lime		Blue	DD - 221	55 - 85
#C0C0C0		#FF0000		#00FF00		#0000FF		CC - 204	44 - 68
	Gray		Purple		Olive		Teal	BB - 187	33 - 51
#808080		#800080		#808000		#008080		AA - 170	22 - 34
	White		Fuchsia		Yellow		Aqua	99 - 153	11 - 17
#FFFFFF		#FF00FF		#FFFF00		#00FFFF		88 - 136	00 - 00

HTML Named Colours

Hex - Dec Conversion

## Annexe 3 : les noms des couleurs HTML

aliceblue	F0F8FF		antiquewhite	FAEBD7		aqua	00FFFF	
aquamarine	7FFFD4		azure	F0FFFF		beige	F5F5DC	
bisque	FFE4C4		black	000000		blanchedalmond	FFEBCD	
blue	0000FF		blueviolet	8A2BE2		brown	A52A2A	
burlywood	DEB887		cadetblue	5F9EA0		chartreuse	7FFF00	
chocolate	D2691E		coral	FF7F50		cornflowerblue	6495ED	
cornsilk	FFF8DC		crimson	DC143C		cyan	00FFFF	
darkblue	00008B		darkcyan	008B8B		darkgoldenrod	B8860B	
darkgray	A9A9A9		darkgreen	006400		darkkhaki	BDB76B	
darkmagenta	8B008B		darkolivegreen	556B2F		darkorange	FF8C00	
darkorchid	9932CC		darkred	8B0000		darksalmon	E9967A	
darkseagreen	8FBC8F		darkslateblue	483D8B		darkslategray	2F4F4F	
darkturquoise	00CED1		darkviolet	9400D3		deeppink	FF1493	
deepskyblue	00BFFF		dimgray	696969		dodgerblue	1E90FF	
firebrick	B22222		floralwhite	FFFAF0		forestgreen	228B22	
fuchsia	FF00FF		gainsboro	DCDCDC		ghostwhite	F8F8FF	
gold	FFD700		goldenrod	DAA520		gray	808080	
green	008000		greenyellow	ADFF2F		honeydew	F0FFF0	
hotpink	FF69B4		indianred	CD5C5C		indigo	4B0082	
ivory	FFFFF0		khaki	F0E68C		lavender	E6E6FA	
lavenderblush	FFF0F5		lawngreen	7CFC00		lightpink	FFB6C1	
lightsalmon	FFA07A		lightseagreen	20B2AA		lightskyblue	87CEFA	
lightslategray	778899		lightsteelblue	B0C4DE		lightyellow	FFFFE0	
lime	00FF00		limegreen	32CD32		linen	FAF0E6	
magenta	FF00FF		maroon	800000		mediamaquamarine	66CDAA	
mediumblue	0000CD		mediumorchid	BA55D3		mediumpurple	9370DB	
mediumseagreen	3CB371		mediumslateblue	7B68EE		mediumspringgreen	00FA9A	
mediumturquoise	48D1CC		mediumvioletred	C71585		midnightblue	191970	
mintcream	F5FFFA		mistyrose	FFE4E1		moccasin	FFE4B5	
navajowhite	FFDEAD		navy	000080		oldlace	FD5E5E	
olive	808000		olivedrab	6B8E23		orange	FFA500	
orangered	FFA500		orchid	DA70D6		palegoldenrod	EEE8AA	
palegreen	98FB98		paleturquoise	AFEEEE		palevioletred	DB7093	
papayawhip	FFEDD5		peachpuff	FFDAB9		peru	CD853F	
pink	FFC0CB		plum	DDA0DD		powderblue	B0E0E6	
purple	800080		red	FF0000		rosybrown	BC8F8F	
royalblue	4169E1		saddlebrown	8B4513		salmon	FA8072	
sandybrown	F4A460		seagreen	2E8B57		seashell	FFF5EE	
sienna	A0522D		silver	C0C0C0		skyblue	87CEEB	
slateblue	6A5ACD		slategray	708090		snow	FFFAFA	
springgreen	00FF7F		steelblue	4682B4		tan	D2B48C	
teal	008080		thistle	D8BFD8		tomato	FF6347	
turquoise	40E0D0		violet	EE82EE		wheat	F5DEB3	
white	FFFFFF		whitesmoke	F5F5F5		yellow	FFFF00	
yellowgreen	9ACD32							

# Ressources

## LIENS :

W3C : <http://www.w3.org>

All HTML : <http://www.allhtml.com>

Alsacr ations : <http://www.alsacreations.com>

## LIVRES :

"M emento XHTML", de Rapha el Goetter (Eyrolles)

"R ussir son site web avec XHTML et CSS", de Mathieu Nebra (Eyrolles)