

Apprendre le HTML

Source : <http://perso.wanadoo.fr/franck.sophie/html32/index.htm>

Didacticiel HTML - Introduction

Ce tutoriel a pour but de vous fournir les clés du HTML, voire un peu plus. Même si vous utilisez un logiciel qui se charge de générer ce code, des connaissances en HTML peuvent s'avérer fort utiles lorsque les choses ne fonctionnent pas comme prévu.

J'ai volontairement choisi la version 3.2 du HTML car elle me paraît plus abordable pour l'apprentissage, et c'est surtout la version la plus largement comprise par les navigateurs aujourd'hui.

Trousse à outils.

Si la plate-forme informatique que vous utilisez n'a guère d'importance (Mac ou PC), il vous faut cependant une "trousse à outils" minimum. Beaucoup de ces outils sont inclus dans votre système (MacOs ou Windows) ou sont disponibles en freeware ou en shareware.

Pour éditer vos fichiers HTML, un simple éditeur de texte suffit : SimpleText, Wordpad ou Write. Pour ceux qui tiennent à utiliser un traitement de texte du style Microsoft Word, il est très important de choisir le format "*Texte seulement*" au moment d'enregistrer vos fichiers.

Pour gérer vos images, des logiciels en shareware tels que Graphic Converter (Mac) ou Paint Shop Pro (PC) font tout à fait l'affaire.

Nommer vos fichiers HTML.

Le nom que vous donnez à vos fichiers HTML est libre, si ce n'est qu'il est limité à 8 caractères sous DOS et Windows 3.x. J'attire seulement votre attention sur la notion de casse, c'est-à-dire le respect des majuscules et minuscules.

En effet, lorsque vous vous trouvez sous système utilisateur (MacOs ou Windows), le fait que vous fassiez appel à un fichier nommé "Bonjour.html" par "bonjour.html" ne pose a priori pas de problème. Tant que vous êtes sous système utilisateur.

Mais sur Internet, la plupart des serveurs - là où seront stockées vos pages personnelles - fonctionnent sous UNIX ou dérivé. Or Unix est sensible à la casse, c'est-à-dire que si vous appelez le fichier "Bonjour.html" en tapant "bonjour.html", vous n'obtiendrez pas de réponse, ou un message d'erreur. Sous Unix, il s'agit de deux fichiers distincts.

En guise de conseil, essayez de n'utiliser que des minuscules dans vos noms de fichier (ou que des majuscules), ce qui vous permettra d'éviter ce risque d'erreur.

L'extension des fichiers HTML.

Lorsque vous enregistrez vos fichiers HTML, il faut leur donner une extension, même si vous travaillez sous MacOs ou Windows 9x! Cette extension est ".html", ou ".htm" si vous travaillez sous DOS ou Windows 3.x.

Règle générale sur les balises HTML.

Les balises HTML fonctionnent par paire : une balise d'ouverture, <TITLE>, et une balise de fermeture, </TITLE>. Comme vous pouvez le voir dans cet exemple (nous en verrons le sens plus tard), le nom de la balise est encadré par les signes "<" et ">". Cela permet au navigateur de différencier le contenu du document du code qui sert à le mettre en forme. La balise de fin reprend le nom de la balise de début mais contient en plus un slash "/", ce qui indique que l'application de la balise cesse à cet endroit.

Comme toute règle, nous verrons par la suite qu'il existe un certain nombre d'exceptions (absence de balise de fin notamment).

Définition et Conventions.

Le langage HTML est un ensemble de codes ou de balises qui permettent de formater, d'afficher et de présenter vos données en vue de leur publication sur Internet.

Chaque balise HTML peut recevoir des attributs qui permettent d'affiner ou de modifier le fonctionnement de la balise. Pour vous permettre de les différencier, j'écrirai les balises en *MAJUSCULES* et les attributs en *minuscules*. Je parle d'*élément* pour définir les balises de début et de fin.

N.B. : Lorsque vous saisissez votre code HTML, sachez que ce langage est insensible à la casse, en d'autres termes, vous pouvez taper vos balises et attributs en majuscules ou minuscules.

Comment procéder?

Le langage HTML est un ensemble de codes ou de balises qui permettent de formater, d'afficher et de présenter vos données en vue de leur publication sur Internet.

Editer vos documents

Il serait sans doute bon que, de temps à autres, vous fassiez des essais avec les balises que je présente. Cela vous permettra de vous rendre compte, réellement, de leur effet et de vous familiariser avec leur utilisation. Voici donc la procédure pour éditer, modifier et vérifier un document HTML.

Créer un nouveau document

1. Pour créer un nouveau document HTML, lancez votre éditeur de texte et ouvrez un nouveau document (si le logiciel ne le fait pas automatiquement).
2. La leçon "Structure générale d'un document HTML" commence par un exemple de code HTML. Sélectionnez-le à la souris dans votre navigateur, copiez-le et collez-le dans votre nouveau document dans votre éditeur de texte.
3. Entre les balises <TITLE> et </TITLE>, remplacez le texte par le titre que vous souhaitez donner à votre document ("Exercices de HTML", par exemple).
4. Dans le menu **Fichier**, choisissez "**Enregistrer sous**". Choisissez l'endroit où vous souhaitez enregistrer ce fichier (l'idéal étant de créer un dossier ou répertoire où vous regrouperez tout vos fichiers d'exercice HTML). Nommez le fichier en prenant soin de lui ajouter l'extension **".html"** ou **".htm"**. Pour ceux qui utilisent un éditeur de texte, n'oubliez pas de choisir le format **"Texte seulement"**. Validez.
5. Retournez dans votre navigateur et ouvrez le fichier que vous venez de créer. Seul le texte "contenu de votre document..." s'affiche. Cependant, en observant la barre de titre de la fenêtre du navigateur, vous pouvez voir que le texte que vous avez entré entre les balises <TITLE> et </TITLE> sert de titre à votre document (pour les non-anglicistes, "title" signifie titre). Pas mal, non? Et ce n'est qu'un début!

Plus tard, pour modifier le contenu de votre fichier HTML, il vous suffit de retourner dans l'éditeur de texte et de le modifier. N'oubliez cependant pas d'enregistrer le fichier avant de le recharger dans votre navigateur.

Charger le fichier dans votre navigateur

Pour que les modifications apportées dans vos fichiers HTML soient prises en compte dans votre navigateur, il vous faut à chaque fois ouvrir le fichier à partir du menu **Fichier**.

Une autre possibilité s'offre à vous : utiliser le bouton "**Rafraîchir**" ou "**Recharger**" de votre navigateur. J'attire cependant votre attention sur le fait qu'en fonction de la configuration de votre logiciel Internet, ce rafraîchissement peut se faire à partir de la version conservée dans la cache. Ainsi, les modifications que vous aurez apporté à votre document HTML n'apparaîtront pas. Pour y remédier, maintenez la touche "**majuscule temporaire**" en cliquant sur le bouton de rechargement.

Vous êtes maintenant prêt pour créer vos propres documents HTML.

Structure d'un document HTML

Voici un exemple de structure générale basique que tout document HTML se doit d'avoir :

```
<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<TITLE>Titre de votre document</TITLE>
<!-- Commentaires personnels non affichés dans le navigateur //-->
</HEAD>
<BODY>
Contenu de votre document qui sera affiché dans le navigateur de
votre visiteur : vous pouvez placer ici du texte, des images, des
sons, des applets java, des scripts ou encore de la vidéo.
<ADDRESS>
Adresse de votre page
Date de création ou de mise à jour
Votre email pour recevoir des commentaires
</ADDRESS>
</BODY>
</HTML>
```

Nous allons maintenant détailler cet exemple et voir la signification des balises qui y sont présentes. Tout document HTML possède deux parties : un en-tête (l'élément <HEAD>) et un corps (l'élément <BODY>). Le tout - en-tête et corps - est encadré par les balises <HTML> et </HTML>. Elles indiquent au navigateur qu'il s'agit de code HTML.

I- L'en-tête du document : <HEAD>

Les éléments contenus dans l'en-tête du document, délimité par les balises <HEAD> et </HEAD>, ne sont pas affichés dans le navigateur Internet, à l'exception du contenu de l'élément <TITLE>.

1- La déclaration de la version de HTML

C'est théoriquement la première ligne de tout document HTML, et elle se trouve avant toute autre déclaration :

```
<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//ENGT;
```

Dans mon exemple, je signale que la version du HTML utilisée est la 3.2, en version anglaise - EN. En fait, il n'existe que la version anglaise.

2- Le titre du document : <TITLE>

Ce sont les balises <TITLE> et </TITLE> qui permettent de donner un titre à vos documents. Le contenu de ces balises apparaît dans la barre de titre de la fenêtre du navigateur.

Cet élément est obligatoire, sans quoi vos pages seront répertoriées sous l'intitulé "Sans titre" ou "Untitled" par certains moteurs de recherche sur Internet.

3- Les commentaires

Tout ce qui se trouve entre les balises `<!--` et `//-->` correspond à un commentaire ou une annotation personnelle qui ne sera pas affiché dans le navigateur. Cet élément peut se placer aussi bien dans l'en-tête ou le corps du document.

Les commentaires peuvent vous servir à séparer logiquement les parties de votre document ou encore à marquer la logique de votre construction. Pensez à utiliser les commentaires. Cela facilite grandement le travail lorsqu'on doit reprendre le fichier, pour une mise à jour par exemple.

II- Le corps du document : <BODY>

C'est là que vous disposez tous les éléments qui seront affichés dans le navigateur de votre visiteur.

Nous verrons par la suite comment implémenter ces éléments, qu'il s'agisse de texte, d'images ou de multimédia.

Laissez votre adresse : <ADDRESS>

Cet élément, situé au début ou à la fin du corps du document, vous permet de laisser vos coordonnées. La manière dont sera rendu le contenu de cet élément dépend directement du navigateur (italique, gras ou autre).

Généralement, on laisse comme coordonnées l'adresse du site, de la page consultée, la date de création ou de modification, et l'e-mail.

Voici un exemple de code :

```
<ADDRESS>  
http://www.keris-graphic.com<BR>  
Mise a jour : sept 2005<BR>  
E-mail : laurent.lescop@keris-graphic.com  
</ADDRESS>
```

Formater le texte

Comme vous avez sans doute pu le constater dans la section sur "*Editer un document HTML*", le texte saisi est affiché au kilomètre, sans mise en forme ni formatage. Pour mieux vous en rendre compte, réduisez ou agrandissez la taille de la fenêtre de votre navigateur. Le texte sera automatiquement réaligné. Pour formater le texte, il faut utiliser du HTML.

Nous allons donc voir comment ajouter du style au texte, et comment formater vos divers paragraphes.

I- Du texte avec du style

1- Les titres

Il existe 6 niveaux de titre en HTML, allant de <H1> à <H6>, <H1> étant le titre le plus important. Leur affichage dépend directement du navigateur que vous utilisez et des polices de caractère choisies.

Dans tous les cas, le texte entre ces balises est placé sur une nouvelle ligne.

Voici un exemple de code :

```
<H1>Mon titre en H1</H1>  
<H2>Mon titre en H2</H2>  
<H3>Mon titre en H3</H3>  
<H4>Mon titre en H4</H4>  
<H5>Mon titre en H5</H5>  
<H6>Mon titre en H6</H6>
```

2- Les différents styles de caractères

Pour appliquer un style à un mot ou à un morceau de phrase, il suffit d'encadrer le ou les mots des balises qui correspondent au style désiré. On peut aussi combiner les styles.

- Le **gras** : Texte - ex. : Voici du **gras**

- L'**italique** : <i>Texte</i> - ex. : Voici de l'*italique*

- Le **monospace** : <TT>Texte</TT> - ex. : voici du texte en police monospace

- **Souligné** : <U>Texte</U> - ex. : voici du texte souligné

- **Barré** : <STRIKE>Texte</STRIKE> ou <S>Texte</S> - ex. : voici du ~~texte barré~~

Pour mêler les styles, il vous suffit d'ajouter les balises les unes à la suite des autres, sans oublier qu'il faudra toutes les fermer:

```
<B><I><U>Voici un exemple de texte</I> partiellement en italique, en gras et souligné</B></U>
```

s'affiche :

Voici un exemple de texte partiellement en italique, en gras et souligné

3-Mettre du texte en évidence et la notion d'accessibilité

Le W3C (World Wide Web Consortium, organisme qui établit entre autres les normes pour le HTML) insiste beaucoup sur le fait de rendre les données accessibles au plus grand nombre, notamment aux personnes souffrant de déficiences visuelles. Ces personnes peuvent utiliser des navigateurs vocaux. Aussi, et pour leur permettre un meilleur accès aux données sur le web, voici les balises qui peuvent les aider à décrire nos pages (le HTML 4 est plus pointu sur ce point, grâce aux feuilles de style). L'affichage et le rendu dépendent de votre navigateur (les rendus peuvent être similaire à de l'italique, du gras ou autre).

- **Mise en évidence simple** : `Texte` - ex.: *Mise en évidence simple*

- **Citation** : `<CITE>Texte</CITE>` - ex.: *Pour faire une citation*

- **Exemple** : `<SAMPLE>Texte</SAMPLE>` - ex.: Ceci est un exemple

- **Variable** : `<VAR>Texte</VAR>` - ex.: *ceci est une variable*

II- Formater le texte

1- Les retours à la ligne

Nous l'avons déjà vu, lorsque vous saisissez votre texte dans votre éditeur de texte, les retours à la ligne que vous faites - via la touche "Entrée" -, ne sont pas affichés dans le navigateur. Et pour cause, il faut faire appel à une balise HTML.

- **Le retour simple** : `
`

Cet élément ne possède pas de balise de fin. Il provoque un retour à la ligne simple à l'endroit où il est placé.

Ex. : `Votre texte
avec un retour à la ligne simple.`

S'affiche ainsi :

Votre texte
avec un retour à la ligne simple.

- **Le retour avec ligne vierge** : `<P>`

Cet élément ne requiert pas non plus de balise de fin. Il affiche un retour à la ligne avec ajout d'une ligne vierge.

Ex. : `Votre texte<P>avec un retour à la ligne et ajout d'une ligne vierge.`

S'affiche ainsi :

Votre texte

avec un retour à la ligne et une ligne vierge.

2- Aligner le texte

En HTML, vous pouvez choisir d'aligner vos paragraphes de texte à gauche (par défaut), au centre ou à droite. Il existe trois solutions, aucune ne permettant cependant de justifier le texte.

- `<P align=left / center / right>Texte</P>` : Oui, il s'agit bien de la même balise que pour l'ajout d'un retour à la ligne, à laquelle on adjoint un attribut, celui de l'alignement (respectivement gauche / centre / droite). A noter aussi, dans cette utilisation de la balise `<P>`, il vous faut utiliser la balise de fermeture pour signaler la fin de l'alignement.

Ex. : `<P align=center>Votre paragraphe est au centre</P>`

s'affiche ainsi :

Votre paragraphe est au centre

- `<CENTER>Texte</CENTER>` : C'est une balise Nestcape, aujourd'hui reconnue par la majorité des navigateurs Internet. Elle centre le paragraphe entre ces balises.

Ex. : `<CENTER>Votre paragraphe est au centre</CENTER>`

s'affiche :

Votre paragraphe est au centre

- `<DIV align=left / center / right>Texte, image, tableau</DIV>` : cette balise permet d'aligner en même temps plusieurs type de données, qu'il s'agisse de texte, d'images ou encore de tableaux. Son fonctionnement est similaire à celui de `<P align=...>`

3- Mettre du texte en retrait : `<BLOCKQUOTE>...</BLOCKQUOTE>`

A l'origine, cet élément devait permettre de différencier un paragraphe de citation du reste du texte, en affichant un retrait à gauche. Mais la majorité des gens utilisent simplement cette balise pour le retrait, sans que le contenu soit forcément une citation. Pour augmenter le retrait, multipliez les blockquote.

Ex. : `<BLOCKQUOTE><BLOCKQUOTE>Texte avec deux retraits</BLOCKQUOTE></BLOCKQUOTE>`

s'affiche ainsi :

Texte avec deux retraits

4- Le texte préformaté : <PRE>.....</PRE>

Le texte que vous saisissez entre ces balises sera affiché tel que vous l'avez entré, c'est-à-dire avec les espaces et les retours à la ligne.

Ex. : <PRE>Voici un texte pre-
formate</PRE>

s'affiche ainsi :

```
Voici un      texte                pre-  
  
formate
```

5- Césure ou pas de césure

- Interdire la césure : <NOBR>Phrase de texte</NOBR> - Le navigateur ne pourra alors pas faire de césure ou de retour automatique à la ligne dans le texte encadré par ces balises. Si besoin est, il ajoutera des ascenseurs horizontaux pour que le lecteur puisse afficher le reste de la ligne.

- Programmer une césure : <WBR>Texte</WBR> - Si la portion de texte pour laquelle vous avez interdit les retours automatiques est longue, il est judicieux de placer cet élément pour permettre au navigateur, si besoin est, d'utiliser la césure programmée pour un retour à la ligne.

6- Les traits de séparation : <HR>

Pour aérer vos textes, il peut être judicieux de séparer les parties par des traits horizontaux. Cet élément HTML ne possède pas de balise de fin, puisque le trait s'affiche sur une nouvelle ligne.

Attributs possibles :

+ align = left / center (défaut) / right - stipule l'alignement du trait.

+ width = pixels / pourcentage - spécifie la largeur du trait, soit en pixels, soit en pourcentage de la taille de la fenêtre.

+ height = pixels - hauteur ou épaisseur du trait en pixels.

+ noshade - efface l'effet d'ombrage affiché par défaut dans le navigateur.

7- Les caractères spéciaux

Vous l'aurez peut-être constaté dans vos essais sur les balises que je vous ai déjà présenté, les caractères accentués de la langue française ne sont pas rendu correctement dans votre navigateur. Il ne s'agit pas d'une erreur. L'anglais, la

langue de base du HTML (vous vous souvenez de la déclaration de la version de HTML utilisé...), ne comporte pas d'accent. Les gens du W3C n'ont cependant pas oublié toutes les autres langues qui sont utilisées sur Terre. Les caractères spéciaux font l'objet de codes qui leur permettent d'être affichés normalement dans un navigateur.

- Les principaux caractères spéciaux pour le français (liste non exhaustive) :

Caractère à afficher	Code en HTML
é	é
è	è
à	à
ù	ù
ê	ê
î	î
ô	ô
â	â
ë	ë
ï	ï
ç	ç
<	<
>	>

Dans votre texte, il vous suffit de remplacer le caractère par le code adéquat.

- Indice et exposant

Dans vos formules mathématiques ou chimiques, il se peut que vous ayez besoin de mettre des caractères en indice ou en exposant.

+ indice : $H_{2}O$

+ Exposant : 10^{3}

Les images

Ne dit-on pas qu'une image vaut bien mille mots! Eh bien nous allons voir comment afficher des images dans vos pages web.

1- Les formats de fichiers graphiques

C'est simple, il n'y en a que deux qui soient reconnues universellement : le GIF et le JPG.

+ **le format GIF** : particulièrement à l'aise avec les aplats de couleurs, ce format permet des fichiers de faible encombrement et permet l'entrelacement (affichage progressif par bandes) et la transparence (format GIF89a).

+ **le format JPG** : il permet de varier le taux de compression et donc de mieux adapter le rapport poids-qualité de l'image. Ce format est recommandé pour les photos et en général les images comportant des dégradés de couleurs.

Il existe d'autres formats, mais il faut alors que votre visiteur ait ajouté à son navigateur les plug-ins nécessaires (petits programmes additionnels).

2- Affichage simple d'une image

Le code de base est des plus simples : ``

Cet élément ne comporte pas de balise de fin.

Ce code suppose cependant que le fichier image se situe dans le même dossier que celui du fichier HTML qui y fait appel. Nous allons donc voir comment faire si ce fichier graphique se situe dans un autre dossier.

3- Suivre le chemin

Indiquer au navigateur le chemin à suivre pour trouver un fichier graphique est sans doute une partie un peu difficile. Mais vous devriez vous en sortir très bien (j'y arrive bien!). Ce que nous allons voir ici pour les images s'applique aussi aux liens vers d'autres fichiers HTML. Pour des raisons pratiques, disons que le dossier/répertoire s'appelle "*images*"

+ l'image se situe dans le dossier/répertoire du fichier HTML :

c'est le cas le plus simple, celui que nous avons vu dans l'exemple précédent.

+ l'image se trouve dans un sous-répertoire/dossier situé dans le dossier où se situe le fichier HTML :

c'est encore relativement simple, le code est ``

Si le dossier "images" se trouve dans un répertoire qui lui-même se situe dans le répertoire du fichier HTML, il suffit d'indiquer le chemin complet : ``

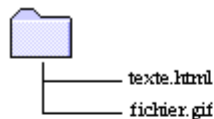
+ l'image se situe dans un répertoire/dossier supérieur à celui du fichier HTML :

on indique alors simplement au navigateur de "remonter" d'un niveau pour trouver le fichier : ``

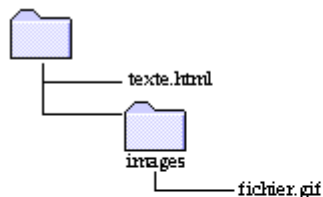
Si le fichier se situe dans un dossier situé à plusieurs niveaux au-dessus de celui du fichier HTML, on multiplie l'indication pour remonter : ``

Si le fichier se situe dans "images", situé plusieurs niveaux au-dessus du fichier HTML : ``

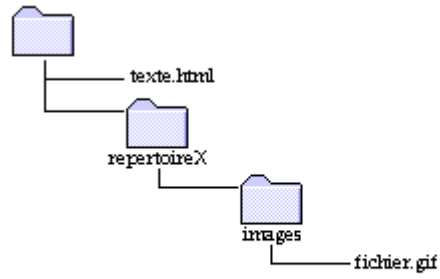
Voici quelques exemples illustrés de chemins :

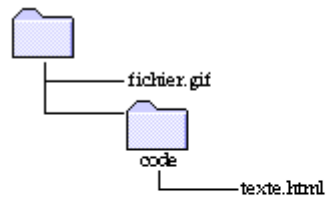


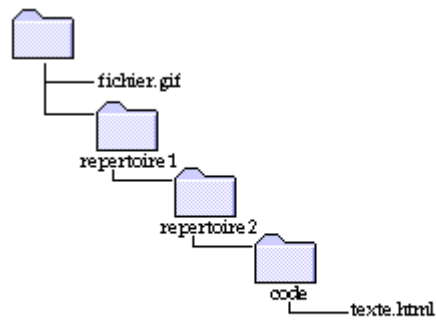
``

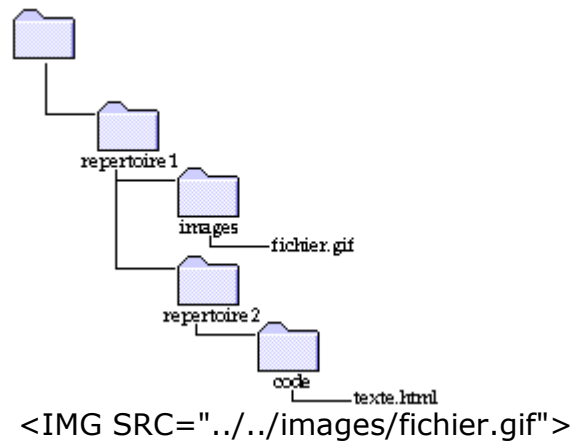


``









4- Les attributs de

- + `width=pixels` - indique la largeur de l'image à afficher
- + `height=pixels` - indique la hauteur de l'image à afficher
- + `alt="texte descriptif"` - quelques mots qui s'affichent avant le chargement de l'image, ou en cas d'impossibilité de charger le fichier
- + `align=left / center / right` - alignement de l'image sur la page
- + `border=pixel` - par défaut, le navigateur affiche un cadre de 1 pixel d'épaisseur autour de l'image. 0 signifie qu'il n'y a pas de cadre.
- + `vspace=pixels` - spécifie un espace vierge en pixels de chaque côtés de l'image
- + `hspace=pixels` - spécifie un espace vierge en pixels en haut et en bas de l'image
- + `lowsrc="fichierbis.jpg"` - spécifie un fichier en basse résolution de l'image à charger. Si votre fichier graphique est lourd, utiliser cet attribut pour faire patienter vos visiteurs avec une image en basse résolution. Cette image doit avoir les mêmes dimensions que l'image finale.

Voici un exemple de code reprenant tous ces attributs :

```
<IMG SRC="Nantes.gif" lowsrc="Nantesbis.jpg" height="125"
width="125" alt="Ma ville" border="0" align=center vspace=10
hspace=10>
```

Ce code affiche une image située dans le même fichier HTML qui l'appelle. Pendant son chargement, il affiche le texte "Ma ville", puis l'image en basse résolution "Nantesbis.jpg". Sa taille est de 125 x 125 et elle est alignée au centre. Elle ne possède pas de bordure et un espace de 10 pixels l'entoure complètement.

5- Une image en fond de page

Nous le verrons plus tard, vous pouvez spécifier la couleur du fond de votre page, celle du texte, et des liens visités. Mais nous allons voir maintenant comment charger une image comme fond de page. Le navigateur répète cette image pour "remplir" entièrement le fond de la fenêtre ouverte.

Un conseil, prenez un fichier très léger et dont les motifs ne nuiront pas à la lecture de vos documents.

Il s'agit en fait d'un attribut de la balise de début `<BODY>` : `<BODY background="fichier.jpg">`

N'oubliez pas de fournir le chemin du fichier si celui-ci ne se trouve pas dans le même dossier/répertoire que le fichier HTML.

Les liens hypertextes

Vous savez maintenant mettre en forme votre texte pour l'afficher sur Internet. Vous savez comment afficher vos images. Mais toute la puissance d'Internet réside sans doute dans sa faculté à relier les sites et les documents entre eux, grâce aux liens hypertextes. Et, cerise sur le gâteau, ce n'est pas plus difficile que ce que nous avons vu jusqu'à présent.

Note : je pars du principe que vous êtes familier de la notion de "chemin" et que vous savez comment indiquer au navigateur l'itinéraire à suivre pour trouver un fichier. Pour un petit rappel, relisez la section concernant le sujet dans la leçon précédente "**Afficher des images**".

1- Créer un lien hypertexte - `<A>`

Un lien HTML possède toujours deux extrémités : le début (le pointeur, ou le lien) et la fin (la cible).

Le début correspond au texte (ou image) qui pointe vers un autre élément. Souvent, il s'agit du texte souligné et de couleur différente au-dessus duquel le curseur de la souris se transforme. Pour une image, si celle-ci ne possède pas de cadre, seule le changement de forme du curseur indique qu'il s'agit d'un lien.

La fin d'un lien est le fichier qui s'affiche lorsque le lien a été cliqué. Il peut s'agir d'un fichier HTML, d'une partie du même fichier (voir la section suivante sur les ancres), d'une image, de vidéo ou de son. Ce document peut se trouver sur le même site (on utilise alors une adresse relative. Ex : `../images/fichier.html`) ou sur un autre site (on utilise alors une adresse absolue. Ex. : `http://www.monsite.com/fichier.html`).

La mise en oeuvre d'un lien hypertexte est très simple : `Lien hypertexte`

Il vous suffit d'indiquer l'adresse (relative ou absolue) du fichier qui devra se charger quand le texte sera cliqué.

2- Utiliser les ancres -

Grâce aux ancres, il est possible de spécifier une section de texte, à l'intérieur d'un document HTML, qui devra être affichée. Pour être plus clair, imaginons que nous ayons un document contenant des poésies, dont l'une des sections, disons la partie 2, concerne les oeuvres de Baudelaire. Si dans un sommaire nous faisons référence à cette partie, au milieu de beaucoup d'autres. Nos visiteurs, intéressés particulièrement par Baudelaire et non par les autres, pourrait grâce à un lien aller directement à cette partie. Il nous faut pour cela construire :

- dans le document contenant les poésies, un marqueur invisible identifiant la partie concernant Le Corbusier :

Insérer une ancre

- A l'endroit que vous souhaitez identifier, placez la balise suivante :
II- L'architecture de Le Corbusier
- Le nom que vous donnez à une ancre doit être unique à l'intérieur d'un même document HTML.
- dans le sommaire, un lien pointant vers le fichier, et plus précisément, à l'intérieur de ce fichier, vers l'ancre identifiant cette partie.

Pointer vers une ancre

- Lorsque vous construisez votre lien vers le fichier archi.html (c'est un exemple!), il vous suffit d'ajouter le nom de l'ancre à la suite du nom du fichier. Comme suit : L'architecture de Le Corbusier
- **N'oubliez surtout pas le signe dièse (#) entre le nom de fichier et celui de l'ancre, car c'est lui qui indique au navigateur de chercher l'ancre "partie2" dans le fichier "archi.html".**

Vous pouvez aussi faire référence à une ancre située dans un même fichier : Le Corbusier

Voilà. Vous pouvez désormais relier vos documents entre eux. Faites des essais, le plus dur étant d'être très rigoureux dans le chemin que l'on fournit.

Couleurs et polices

I- Définition des couleurs en HTML

Il existe deux possibilités en HTML pour définir une couleur :

- l'utilisation des 16 couleurs standard (héritage des premiers écrans VGA sous Windows)
- l'utilisation de codes hexadécimaux

1- Les couleurs standard

Elles sont donc au nombre de 16, et sont reconnues par tous les systèmes. En voici la liste :

Aqua	black	blue	fuchsia
gray	green	lime	maroon
navy	olive	purple	red
silver	teal	white	yellow

Leur emploi est simple, comme nous le verrons ultérieurement.

2- Les codes hexadécimaux

Cela se complique un peu. Pour être plus en harmonie avec le langage système, et pour permettre l'utilisation de toutes les couleurs (16 couleurs, c'est vraiment court!), les créateurs du HTML ont eu l'idée de convertir les composantes RVB de chaque couleur en un code hexadécimal (base 16 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f). Le mieux, si vos outils graphiques ne font pas la conversion, c'est de surfer sur le web à la recherche d'un shareware ou d'un freeware qui se chargera de la conversion pour vous. Vous n'avez qu'à cliquer dans une palette pour choisir une couleur, et le code vous est fourni.

Pour utiliser ces couleurs converties en code hexadécimaux, il vous faut précéder la valeur par un signe dièse (#).

Voici quelques exemples de codes hexadécimaux de couleurs :

code hexadécimal pour le blanc : `FFFFFF`

code hexadécimal pour le noir : `000000`

II- Les éléments de la page

Nous avons vu dans la partie "**Afficher des images**" qu'il était possible de spécifier un fichier graphique qui sert de fond de page. Il est cependant aussi possible d'intervenir sur la couleur d'autres éléments de la page, tel que le fond de la page, le texte, les liens, les liens visités ou encore le lien cliqué. Tout ceci se modifie en ajoutant des attributs à la balise `<BODY>`.

Les attributs de <BODY> :

+ bgcolor= "couleur / code hexadécimal" - spécifie la couleur du fond de page.

+ text="couleur / code hexadécimal" - spécifie la couleur par défaut pour tout le texte du document.

+ link= "couleur / code hexadécimal" - spécifie la couleur des liens hypertextes et du cadre des images agissant comme des liens hypertextes.

+ vlink= "couleur / code hexadécimal" - spécifie la couleur des liens visités dans le document.

+ alink= "couleur / code hexadécimal" - spécifie la couleur du lien au moment où il est cliqué (bouton de la souris en bas).

Voici un exemple de code :

```
<BODY bgcolor="#000000" text="white" link="blue" vlink="yellow">
```

Le fond de page sera noir, le texte blanc, les liens en bleu et les liens visités seront jaunes.

Note : Lorsque vous spécifiez une couleur par son code hexadécimal, n'oubliez pas le signe dièse (#), sans lequel votre code ne sera pas reconnu.

III- Les polices de caractères

Grâce à la balise , il vous est possible de spécifier plusieurs choses pour une section de texte :

- la police de caractère
- la taille des caractères
- la couleur du texte

1- La balise

Les attributs de la balise ...

+ face="nom de la police1, nom de la police2, etc." - spécifie une liste de polices de caractère par ordre de préférence. Si la police n'est pas présente dans le système de votre visiteur, le navigateur cherche la suivante et ainsi de suite. Si aucune n'est présente, il affiche le texte dans la police de caractère par défaut.

+ size=valeur absolue / valeur relative - spécifie la taille des caractères. Deux méthodes coexistent :

- la valeur absolue : les tailles vont de 1 à 7, 3 étant la taille normale par défaut. On spécifie alors la taille par `size=5`, ce qui a pour effet d'agrandir de deux tailles les caractères ainsi encodés.

- la valeur relative : 3 étant la taille standard, on spécifie alors `size=+2`, ce qui a le même effet que dans l'exemple précédent.

+ `color="couleur standard / code hexadécimal"` - spécifie la couleur du texte encodés.

Voici un exemple de code :

```
<FONT face="Helvetica, Times" size=+2 color="#AE3F1A">Mon texte special</FONT>
```

Ce qui s'affiche ainsi : **Mon texte spécial**

2- Changer la taille par défaut

Pour une raison quelconque, vous pouvez désirer changer la taille par défaut du texte. La balise `<BASEFONT>` (pas de balise de fin) a été conçue à cet effet.

Placée à l'endroit de votre choix, elle stipule une taille de caractère par une valeur absolue ou relative (voir plus haut) à tout le texte qui la suit. Si vous faites appel à la balise `` pour modifier la taille d'une portion de texte, la taille normale sera celle spécifiée dans `<BASEFONT>`. Ceci est particulièrement important pour utiliser des valeurs relatives.

Ne possédant pas de balise de fin, si vous désirez retrouver la taille par défaut du HTML (c'est-à-dire 3 en valeur absolue), il vous faudra disposer une nouvelle balise `<BASEFONT>` pour l'indiquer.

Exemple de code :

```
<BASEFONT size=+2>
```

Le texte qui suit cette balise verra sa taille augmentée de deux niveaux.

Les tableaux

Nous entamons ici une partie un peu plus ardue. Pas par sa complexité, mais par le nombre de balises et d'attributs que nous allons voir. Mais cela en vaut la peine. Tout d'abord, un tableau permet de structurer la présentation des données, mais surtout - et comme bon nombre d'internautes l'ont compris- les tableaux permettent de palier les lacunes du HTML en matière de mise en page : il devient possible d'aligner plusieurs lignes de texte à côté d'une image, de présenter le texte en colonnes, d'aligner correctement plusieurs images, d'imbriquer des tableaux dans un tableau, et j'en passe.

Nous allons donc voir dans un premier temps la balise `<TABLE>` et ses attributs, puis les différentes balises de cellule. Enfin, un exemple de code pour un tableau vous permettra de visualiser la structure de l'ensemble.

I- Implémenter un tableau - `<TABLE>`

Les balises `<TABLE>` et `</TABLE>` marquent le début et la fin d'un tableau.

Il est possible d'ajouter un certain nombre d'attribut à la balise de début pour modifier l'aspect du tableau.

Les attributs de `<TABLE>` :

+ `b bgcolor="couleur standard / code hexadécimal"` - spécifie la couleur de fond du tableau.

+ `background="fichier graphique"` - spécifie un fichier image en guise de fond pour le tableau. Les formats GIF et JPG sont reconnus universellement.

+ `border="pixel"` - spécifie l'épaisseur en pixels du cadre et des bordures intérieures du tableau ("0" signifie qu'il n'y a pas de bordure, et 1 est la valeur par défaut).

+ `bordercolor="couleur standard / code hexadécimal"` - spécifie la couleur du cadre et des bordures intérieures du tableau.

+ `height="pixels"` - spécifie la hauteur en pixels du tableau.

+ `width="pixels / %"` - spécifie la largeur du tableau, soit en pixels, soit en pourcentage par rapport à la taille de la fenêtre.

+ `cellpadding="pixels"` - spécifie l'épaisseur des bordures intérieures du tableau.

+ `cellspacing="pixels"` - spécifie l'espace en pixels entre le contenu des cellules et le bord de la cellule.

Exemple de code :

```
<TABLE border="0" align="center" bgcolor="gray" width="50%"  
cellpadding="0" cellspacing="2">
```

Ici, le tableau n'aura pas de bordure, aura un fond gris, une largeur égale à la moitié de celle de la fenêtre du navigateur. L'épaisseur des bordures internes sera nulle et l'espace entre le bord des cellules et leur contenu sera de 2 pixels.

II- Les cellules d'un tableau

1- Les titres

Un tableau se lit et se construit de haut en bas, en suivant chaque ligne de gauche à droite.

`<CAPTION>` Titre du tableau`</CAPTION>` : fournit un titre au tableau.

- `<TH>` Colonne 1 `</TH>` : utilisées à la place des balises de cellule (voir plus bas), elles permettent de donner un titre à une colonne ou à une rangée d'un tableau.

Attributs de `<TH>` :

+ `align="left (défaut) / center / right"` - spécifie l'alignement des données à l'intérieur de la cellule (gauche, centre ou droite).

+ `valign="top / middle (défaut) /bottom"` - spécifie l'alignement vertical des données à l'intérieur de la cellule (en haut, au milieu ou en bas).

+ `bgcolor="couleur standard / code hexadécimal"` - spécifie la couleur de fond de la cellule.

+ `background="fichier graphique"` - spécifie un fichier image en guise de fond pour la cellule. Les formats GIF et JPG sont reconnus universellement.

+ `height="pixels"` - spécifie la hauteur de la cellule de titre.

+ `width="pixels / %"` - spécifie la largeur de la cellule de titre (% par rapport à la largeur totale du tableau).

2- Les cellules

+ Pour créer une **nouvelle ligne** dans le tableau (y compris pour les titres de colonnes utilisant les balises `<TH>` : `<TR>`vos cellules `</TR>`

Attributs de `<TR>` :

+ `align="left (défaut) / center / right"` - spécifie l'alignement des données à l'intérieur des cellules de la rangée (gauche, centre ou droite).

+ `valign="top / middle (défaut) /bottom"` - spécifie l'alignement vertical des données à l'intérieur des cellules de la ligne (en haut, au milieu ou en bas).

+ `bgcolor="couleur standard / code hexadécimal"` - spécifie la couleur de fond des cellules de la ligne.

+ `background="fichier graphique"` - spécifie un fichier image en guise de fond pour les cellules de la ligne. Les formats GIF et JPG sont reconnus universellement.

+ `height="pixels"` - spécifie la hauteur de la ligne.

+ Pour **créer une cellule** : `<TD>` Le contenu de la cellule (texte, images,...)
`</TD>`

Attributs de `<TD>` :

+ `align="left (défaut) / center / right"` - spécifie l'alignement des données à l'intérieur de la cellule (gauche, centre ou droite).

+ `valign="top / middle (défaut) /bottom"` - spécifie l'alignement vertical des données à l'intérieur de la cellule (en haut, au milieu ou en bas).

+ `bgcolor="couleur standard / code hexadécimal"` - spécifie la couleur de fond de la cellule.

+ `background="fichier graphique"` - spécifie un fichier image en guise de fond pour la cellule. Les formats GIF et JPG sont reconnus universellement.

+ `height="pixels"` - spécifie la hauteur de la cellule.

+ `width="pixels"` - spécifie la largeur de la cellule.

+ `colspan="nombre de colonnes"` - spécifie que la cellule occupe plusieurs colonnes (vers la droite).

+ `rowspan="nombre de lignes"` - spécifie le nombre de lignes occupées par la cellule (vers le bas).

III- Exemple de code pour un tableau

```
<TABLE width=400 border=1 bordercolor="#000000" cellpadding=2
cellspacing=2 bgcolor="gray">
<TR>
<TD align=center width=150> cellule 1 de la ligne 1</TD>
<TD align=center width=250>cellule 2 de la ligne 1</TD>
</TR>
<TR>
<TD align=center colspan=2 bgcolor="yellow"> cellule unique de la
```

```

ligne 2</TD>
</TR>
<TR>
<TD>cellule 1 de la ligne 3</TD>
<TD rowspan=2 bgcolor="red">cellule 2 des lignes 3 et 4</TD>
</TR>
<TR>
<TD>cellule 1 de la ligne 4</TD>
</TR>
</TABLE>

```

Ce qui devrait apparaître comme suit :

cellule 1 de la ligne 1	cellule 2 de la ligne 1
cellule unique de la ligne 2	
cellule 1 de la ligne 3	cellule 2 des lignes 3 et 4
cellule 1 de la ligne 4	

Faites des essais, vous prendrez mieux conscience de l'utilisation des balises pour les tableaux. Souvenez-vous qu'il est plus facile de faire quelque chose quand on sait à quoi doit ressembler le résultat final. Aussi n'hésitez pas dans un premier temps à crayonner la forme de votre tableau, à prévoir le nombre de lignes, le nombre de colonnes etc.

Les frames

Les frames sont les cadres créés dans la fenêtre de votre navigateur. Chaque cadre peut recevoir un fichier HTML différent. Ce principe est surtout utilisé pour des barres de navigations, des hauts de page (où sont placés des publicités ou plus simplement le nom de la compagnie ou du site visité). Un moment très en vogue, les frames tendent à se faire plus rares sur Internet. D'abord parce que la mise à jour du site est plus difficile (vous le verrez dans cette leçon) et surtout parce que les frames accroissent sensiblement les temps de chargement pour vos visiteurs. Qui plus est, esthétiquement, les frames ont tendances à "charger" le design des sites.

Dans le cas où nous construisons une page contenant deux frames, il nous faut au moins trois fichiers (il faut néanmoins ajouter un fichier par cadre créé - son contenu) :

- le fichier <FRAMESET>, celui qui répartit les cadres dans la fenêtre
- deux fichiers HTML pour le contenu de chaque cadre

1- Le fichier <FRAMESET>

Comme un tableau, un fichier frameset se construit en lisant les cadres de haut en bas, et de gauche à droite. Mais commençons par un exemple de code simple :

```
<HTML>
<HEAD>
<TITLE>Mon site</TITLE>
</HEAD>
<FRAMESET cols="20, *">
<FRAME SRC="fichier1.html" name="colGauche">
<FRAME SRC="fichier2.html" name="colDroite">
</NOFRAMES>
```

Ce site a été construit avec des frames. Il semble que votre navigateur ne supporte pas ces balises. Voici un lien pour visiter notre site sans frames : Site sans frames

```
</NOFRAMES>
</FRAMESET>
</HTML>
```

Le fichier <FRAMESET> est le fichier qui crée et répartit dans la fenêtre les différents cadres (frames) de notre page. Les balises <FRAMESET> et </FRAMESET> remplacent les balises traditionnelles <BODY> et </BODY>.

Ici, nous avons créé deux colonnes, la première fait 20 pixels de large et la seconde occupe le reste disponible de la fenêtre du navigateur. Dans la colonne de gauche de 20 pixels de large - son nom étant "colGauche" -, nous chargeons le fichier "fichier1.html" et dans la colonne de gauche - "colGauche" -, nous chargeons le fichier "fichier2.html".

Au cas où le navigateur de notre visiteur ne supporterait pas les balises de frames, nous fournissons un contenu alternatif via les balises `<NOFRAMES>`. Ici, un lien vers [fichier2.html](#).

2- Les attributs pour le fichier `<FRAMESET>`

Les attributs de la balise `<FRAMESET>` :

+ `cols="pixels / %, pixels / %"` - crée des colonnes dont la largeur peut être spécifiée en pixels ou en pourcentage de la fenêtre totale. Les dimensions pour chaque colonne sont séparées par des virgules. (on note aussi qu'il est possible d'indiquer le signe "*" pour indiquer que la colonne occupe l'espace restant).

+ `rows="pixels / %, pixels / %"` - crée des cadres horizontaux (lignes). **Il n'est pas possible d'utiliser cols et rows dans le même frameset** (voir dans l'exemple en fin de leçon).

+ `framespacing="pixels"` - indique l'espace entre les deux cadres.

+ `frameborder="pixels"` - épaisseur de la bordure des cadres.

+ `bordercolor="code hexadécimal"` - couleur de la bordure des cadres (ici, les noms des 16 couleurs standard ne sont pas possibles).

+ `border="pixels"` - épaisseur de la bordure entre les cadres.

NOTE IMPORTANTE : si vous désirez que vos cadres n'aient pas de bordure, il vous faut fixer les trois attributs suivants à la valeur de "0" pour que cela fonctionne sous Netscape **et** Internet Explorer : `framespacing`, `frameborder` **et** `border`.

Certains de ces attributs sont redondants, mais c'est parce qu'ils ne fonctionnent soit que sous Internet Explorer soit sous Netscape.

Les attributs des balises `<FRAME>` :

+ `src="fichier"` - indique le fichier initialement chargé dans la fenêtre.

+ `name="texte"` - spécifie un nom pour la fenêtre créée. C'est particulièrement important car vous ferez appel à ce nom lorsque vous voudrez charger un fichier dans ladite fenêtre.

Bien que le nom que vous donnez à vos fenêtres soit libre, il existe certains noms qui sont réservés en HTML :

- "`_top`" : le fichier est chargé dans la fenêtre entière (efface les frames)

- "`_blank`" : une nouvelle fenêtre est ouverte dans votre navigateur pour y charger le fichier. L'ancienne fenêtre reste ouverte en arrière plan.

- `"_parent"` : ouvre le fichier dans une fenêtre supérieure hiérarchiquement (dans le cas où plusieurs fenêtres du navigateur sont ouvertes).

+ `scrolling="yes / no / auto"` - autorise ou non la présence d'ascenseurs de défilement pour le cadre. "auto" laisse au navigateur le soin d'afficher des ascenseurs si besoin est (c'est l'option par défaut si rien n'est spécifié).

+ `marginheight="pixels"` - marges disposées en haut et en bas du cadre.

+ `marginwidth="pixels"` - marges disposées de chaque côté du cadre.

3- Des fichiers HTML et des frames

En fait, rien ne différencie à première vue les fichiers HTML pour un site avec des frames des autres fichiers HTML. La seule différence s'observe dans les liens hypertextes : il faut préciser la frame de destination pour l'affichage du fichier à charger. Par défaut, si rien n'est indiqué, le fichier se charge dans le cadre où se trouvait le lien hypertexte.

Voici un exemple de code :

```
<A HREF="monExemple.html" target="colDroite">Un exemple de chargement dans une frame</A>
```

En guise de rappel, trois noms réservés vous permettent de sortir des frames : `"_top"`, `"_parent"`, `"_blank"` (à utiliser avec l'attribut `target`).

4-Exemple de fichier frameset

Nous allons construire trois frames : deux colonnes, dont celle de gauche contenant deux rangées.

```
<HTML>
<HEAD>
<TITLE>Mon site en frames</TITLE>
</HEAD>
<FRAMESET cols="150, *" border="5" frameborder="5"
bordercolor="#000000">
<FRAMESET rows="50%, 50%">
<FRAME SRC="fichier1.html" name="hautGauche">
<FRAME SRC="fichier2.html" name="basGauche">
</FRAMESET>
<FRAME SRC="fichier3.html" name="colDroite">
<NOFRAMES>
```

```
Votre navigateur ne supporte pas les frames. Voici un acces a <A
HREF="fichier3.html">notre site sans frames</A>
</NOFRAMES>
</FRAMESET>
</HTML>
```

Voici deux captures d'écran qui vous permettront de visualiser le résultat. On note que sous Explorer 3, la couleur et l'épaisseur des bordures de cadre ne fonctionnent pas.



Capture du résultat sous Internet Explorer 3



Capture du résultat sous Netscape 4

Comme d'habitude, faites des essais, en commençant par le plus simple (en reprenant les exemples que je donne), puis incorporez progressivement les divers attributs. A chaque fois, vérifiez le résultat dans votre navigateur.

Récapitulatif des balises

Chose promise, chose due. Voici un récapitulatif des balises présentées dans cette petite formation au HTML 3.2.

Balises	Attributs	Commentaires
Structure du document HTML		
<doctype HTML Public "-//W3C//DTD HTML 3.2//EN">		Déclaration de la version de HTML utilisée dans le document
<HTML> ... </HTML>		
<HEAD> ... </HEAD>		En-tête d'un document HTML
<BODY> ... </BODY>	background="fichier graphique" bgcolor="couleur ou code hexadécimal" text="couleur ou code hexadécimal" link="couleur ou code hexadécimal" vlink="couleur ou code hexadécimal" alink="couleur ou code hexadécimal"	Corps du document HTML
<!-- ... //-->		Commentaires personnels
<ADDRESS> ... </ADDRESS>		Vos coordonnées
<META>	http-equiv="refresh" type="..." content="..."	Informations sur le document et/ou son auteur
Justification		
<P>		retour à la ligne plus une ligne vierge
<P> ... </P>	align="left / center / right"	alignement du paragraphe
 		retour à la ligne simple
<CENTER>...</CENTER>		centrer
<DIV> ... </DIV>	align="left / center / right"	aligner plusieurs éléments en même temps
<HR>	align="left / center / right" width="pixels ou %" height="pixels" noshade	trait horizontal de séparation
<BLOCKQUOTE>... </BLOCKQUOTE>		Retrait de paragraphe
<PRE>...</PRE>		le texte est affiché tel que saisi dans l'éditeur de texte
Style de caractère		
<Hn>...<Hn>		Titres, avec des niveaux allant de 1 à

		6 (croissant en importance)
...		gras
<I>...</I>		italique
<U>...</U>		souligné
<TT>...</TT>		police monospace (type Courier)
<STRIKE>...</STRIKE>		barré
<S>...</S>		barré
Mise en évidence		
<XMP>...</XMP>		code HTML
<DFN>...</DEF>		définition
<VAR>...</VAR>		variable
<SAMPLE>...</SAMPLE>		exemple
<CITE>...</CITE>		citation
...		mise en évidence
Les listes		
...	type="square / circle / disc"	liste à puces
...	type="1 / A / a / I / i" start="n"	liste numérotée
		élément de liste (à puces ou numérotée, dir ou menu)
<DL>... </DL>		liste descriptive
<DT>		titre de rubrique dans une liste descriptive
<DD>		élément de rubrique dans une liste descriptive
<MENU>...</MENU>		liste en présentation condensée
<DIR>...</DIR>		liste sous forme de listing
Les images		
	src="chemin/fichier_graphique" align="left / center / right" height="pixels" width="pixels" lowsrc="fichierBis" alt="description" border="pixels" vspace="pixels" hspace="pixels" usemap="nom_map"	
Les liens hypertextes		
<A>...	href="fichier_cible" target="fenetre_cible" (noms réservés : "_top, _blank, _parent") name="nom_de_l'ancre"	
Les polices de caractères		

...	face="police1, police2, ..." size="+n ou valeur absolue (1 à 7)"	sélectionne la police de caractère et sa taille
<BASEFONT>	size="valeur relative ou valeur absolue"	spécifie la taille par défaut de la police pour le texte qui suit
^{...}		caractères en exposant
_{...}		caractères en indice
Les couleurs		
	aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow	les 16 couleurs standard
	#code hexadécimal	codes hexadécimaux permettant de couvrir toutes les couleurs
Les tableaux		
<TABLE>....</TABLE>	bgcolor="couleur ou code hexadécimal" background="fichier graphique" border="pixels" bordercolor="couleur ou code hexadécimal" align="left / center / right" height="pixels" width="pixels ou %" cellpadding="pixels" cellspacing="pixels"	Marque les limites d'un tableau
<CAPTION>...</CAPTION>		titre pour le tableau
<TH>...</TH>	align="left / center / right" valign="top / middle / bottom" width="pixels / %" height="pixels / %" bgcolor="couleur / code hexadécimal" background="fichier graphique"	Cellule de titre pour une rangée ou une colonne de tableau
<TR>...</TR>	align="left / center / right" valign="top / middle / bottom" width="pixels / %" height="pixels / %" bgcolor="couleur / code hexadécimal" background="fichier graphique"	construit une ligne de tableau
<TD>...</TD>	align="left / center / right" valign="top / middle / bottom" width="pixels / %" height="pixels / %" bgcolor="couleur / code hexadécimal" background="fichier graphique"	construit une cellule de tableau
Les frames		
<FRAMESET>.....</FRAMESET>	cols="pixels / % / *" rows="pixels / % / *" framespacing="pixels" frameborder="pixels"	Vient en lieu et place de l'élément <BODY> dans un fichier frameset.

	border="pixels" bordercolor="code hexadécimal" marginheight="pixels" marginwidth="pixels" scrolling="yes / no / auto"	
<FRAME>...</FRAME>	src="fichier" name="nom_de_fenêtre"	définit chaque cadre/frame
<NOFRAMES>...</NOFRAMES>		contenu alternatif pour les navigateurs ne supportant pas les frames
Les images-maps		
<MAP>...</MAP>	name="nom_du_map"	délimite la définition des maps (zones d'activité de l'image)
<AREA>	shape="rect / circle / polygon" coords="x1, y1, x2, y2" href="fichier_cible_de_la_zone"	définition d'une zone active. Voici les coordonnées à saisir pour chaque forme : + rect : coin supérieur gauche et coin inférieur droit + circle : point à gauche, point en haut, rayon + polygon : chaque point

Les moteurs de recherche

Vous avez construit votre site Internet. Mais voilà, il vous faut maintenant le faire connaître. Le travail est en partie fait sur certains moteurs de recherche. Ceux-ci utilisent en effet les balises META que l'on place dans l'en-tête du document HTML. Pour d'autres, il faut s'inscrire.

1- Les balises META

Placées dans l'en-tête de vos documents HTML, ces balises permettent aux créateurs de sites Internet de fournir un certain nombre d'informations sur le site, la page ou le créateur lui-même.

Il n'existe pas réellement de règles en la matière. On place généralement ces balises après <TITLE> ... </TITLE>. Pour ce qui est du contenu, voici quelques balises régulièrement rencontrées.

+ `<META HTTP-EQUIV="refresh" CONTENT="5; http://fichier_a_charger">`
Cette balise permet de charger automatiquement un deuxième fichier ("fichier_a_charger"> au bout de 5 secondes.

+ `<META TYPE="author" CONTENT="fra.so">`
Indique que l'auteur de la page est fra.so

+ `<META TYPE="keywords" CONTENT="art, html, ...">`
Indique une liste de mots-clés correspondant au contenu de la page (éviter les accents). Ces mots-clés sont susceptibles d'être les critères entrés dans un moteur de recherche.

+ `<META TYPE="lang" CONTENT="fr, en, us-en, span">`
Indique la ou les langues accessibles sur le site.

+ `<META TYPE="description" CONTENT="breve description par des phrases du contenu du site">`
Petite description du contenu du site.

Il existe beaucoup d'autres balises META. En demandant l'affichage des sources des documents que vous consultez, vous en rencontrerez sûrement. Si vous comprenez l'anglais, leur sens est souvent évident.