



Qu'est-ce qu'est le HTML ?

Hyper Text Markup Language ou langage hypertexte est le langage à partir duquel sont créées les pages du web. Un site web est constitué d'un ou de plusieurs documents HTML, appelées aussi pages. Pour se déplacer d'une page à une autre on passe par l'intermédiaire d'hyperliens. Un document HTML est un document qui a l'extension HTML ou HTM indifféremment. C'est un simple fichier texte avec l'extension htm ou html.

Le HTML n'est pas un langage de programmation comme le C ou le C++. Il existe sur le web des langages autres que le HTML. C'est le cas des langages dynamiques comme le JavaScript (.js), le perl (.pl ou .cgi), le PHP (.php ou .php3) ou l'ASP (.asp). Certains de ces langages vont s'interpréter dans le navigateur, exemple du JavaScript et d'autres sur le serveur comme c'est le cas pour les autres cités précédemment. Certains de ces langages vont d'ailleurs générer des pages HTML statiques.

Le langage HTML est un ensemble de règles qui indiquent à un navigateur (IE, Netscape par exemple) comment afficher une page sur le navigateur. Un groupe, le consortium WWW ou W3C se charge de normaliser ce langage en précisant les directives. La norme actuelle s'appelle HTML 4 (tendance HTML 5).

Chapitre 1 : Concepts de bases du HTML

Une page web est un fichier texte qui est interprété par votre navigateur. Ce fichier est composé de deux types d'éléments : le contenu et les indications pour l'affichage du contenu. Ces dernières sont nommées balises. Pour être reconnaissable par le navigateur, elles sont entourées des signes `<` `>`

exemple : `<body>` Leur but étant d'indiquer la manière dont les données doivent être affichées, elles entourent le texte à visualiser.

On nomme balise entrante celle qui indique le début du formatage et balise sortante celle qui indique la fin du formatage. La balise sortante se distingue par le signe `/` qui se trouve juste après `<` comme par exemple `Bonjour` qui donne Bonjour.

A toute balise entrante doit correspondre une balise sortante, sauf pour quelques rares exceptions. Le navigateur appliquera sur le reste de votre page la commande de la balise entrante, s'il ne trouve pas de balise sortante. Tout cela semble bien compliqué à lire, mais dans la pratique vous l'appliquerez sans même vous en rendre compte.

1. Les navigateurs

Comme je l'ai dit plus haut, votre page web est un fichier texte qui sera interprété par le navigateur qui va la lire. C'est ici que commencent les difficultés. Il faut savoir que, même si des règles très précises, sont établies par un organisme international nommé le W3C, les navigateurs ne les suivent pas toujours. Ce qui implique que votre page peut avoir un aspect différent sur Netscape et sur Internet Explorer. En HTML, vous n'avez pas la maîtrise totale de votre document.

Les outils dont vous avez besoin

Tout ce dont vous avez besoin pour créer votre première page HTML est dans votre ordinateur. Il vous faut un éditeur de texte tel que le Bloc -notes ou Notepad de Windows.

Attention, ne prenez pas un traitement de texte, car ce dernier va introduire un formatage de texte complètement illisible par le navigateur.

Vous avez besoin d'un navigateur. Lorsque vous maîtriserez bien le HTML, vous réaliserez que vous avez besoin de plusieurs navigateurs pour contrôler l'affichage de votre page. Il vous est bien évidemment possible d'utiliser un éditeur **WYS WYG** tel que **Dreamweaver** ou Front Page. Je ne veux pas lancer ici la bataille entre les puristes qui tapent leur texte dans un éditeur de texte et ceux qui n'utilisent que des éditeurs WYSWYG. Chacun utilise l'instrument qui lui convient le mieux.

Si vous êtes sous Windows, la méthode la plus simple et la moins chère consiste à taper votre texte dans Notepad, d'enregistrer votre texte avec l'extension .htm (et non pas .txt) et de voir le résultat en le visionnant dans Internet Explorer. (je demande pardon aux utilisateurs Linux et Mac... mais ils sauront trouver les outils correspondants)

Un document HTML est délimité par la balise `<html></html>`. Il se compose de deux parties :

- **L'entête** qui contient des informations destinées au navigateur tel que le titre, les styles ou les balises de référencement. Cette partie, à part le titre, est invisible au visiteur de la page. Elle est délimitée par la balise `<head></head>`
- **le corps de la page** qui est délimité par la balise `<body></body>`. Il contient tout ce que vous désirez transmettre à votre visiteur.

Votre première page en HTML peut se présenter comme ceci :

```
<html>
  <head>
    <title>Ma première page</title>
  </head>
  <body>
    Bonjour
  </body>
</html>
```

Quelques explications sur vos premières balises

- `<html>` Début du document de type HTML.
- `<head>` Début de l'entête
- `<title>` Début du titre de la page.
- `</title>` Fin du titre de la page.
- `</head>` Fin de la zone d'en-tête.
- `<body>` Début du corps.
- `</body>` Fin du corps.
- `</html>` Fin du document de type HTML.

Les balises ne sont pas "case sensitive". Il est donc possible d'écrire `<HTML>`, `<html>`, `<Html>`, etc.

Dans l'exemple, les lignes sont incrémentées ou décrémentées selon que l'on rencontre une balise entrante ou sortante. C'est une bonne habitude à prendre.

Lorsque les pages deviennent plus grandes, il est plus facile de se retrouver dans la source du document et d'y trouver, par exemple, la balise que n'a pas été refermée par oubli.

Vous constaterez que le nom de la page est affiché dans la barre du navigateur et que Bonjour apparaît sur la page.

2. Syntaxe d'une balise HTML

Comme on l'a défini ci-haut, une balise est un mot clé du langage HTML compris entre les caractères "<" et ">". Ce mot clé est peut être précédé de caractère /.

Il s'agit dans ce cas d'une balise de fermeture qui indique la fin des actions d'une autre balise du même mot clé appelée balise d'ouverture. Toute paire de balises ouverture/fermeture est appelée balise conteneur, exemple `<body></body>`.

Il existe des balises qui ne possèdent pas de correspondance en fermeture, on appelle ces balises ponctuelles, exemple `
`.

Une balise peut posséder une ou plusieurs caractéristiques définies par des attributs. Un attribut possède une valeur ou non.

Syntaxes de balise:

- Cas d'une balise conteneur

```
<nombalise attribut1="val1" attribut2="val2" attribut3> </nombalise>
```

- Cas d'une balise ponctuelle

```
<nombalise attribut1="val1" attribut2="val2" attribut3 >
```

La balise `<body>`. Cette balise détermine le corps du document. C'est dans cette balise qu'on intègre les éléments d'une page web, donc les titres, textes, images, tableaux, liens etc...

3. Les attributs de <body>:

- **bgcolor** : Définit la couleur du fond de l'écran
- **text** : Définit la couleur du texte
- **link** : Définit la couleur des liens
- **vlink** : Définit la couleur des liens déjà visités
- **alink** : Définit la couleur des liens activés, c'est à dire quand le pointeur de la souris passe dessus
- **background** : Définit l'image (gif ou jpeg) à utiliser comme fond d'écran
- **bgproperties** : Si définie comme "fixed", l'image de fond ne défile pas (uniquement avec I.Explorer)
- **leftmargin** : Définit la largeur de la marge de gauche en pixels
- **topmargin** : Définit la largeur de la marge du haut en pixels

Exemples:

Page avec couleur de fond en blanc (couleur par défaut)

```
<body bgcolor="white" text="black" link="blue" vlink="green" alink="red">
```

Page avec image de fond référencée par le fichier "images/fond.jpg"

```
<body background="images/fond.jpg" text="black" link="blue" vlink="green" alink="red">
```

4. Les commentaires dans un document HTML

Vous pouvez commenter votre code source sans que le navigateur affiche ce commentaire. Cela peut s'avérer pratique, si vous êtes plusieurs à travailler sur le même document. Cette instruction est aussi utilisée lors de l'écriture d'un script JavaScript pour cacher le code aux navigateurs non compatibles.

Syntaxe : `<!.Texte à commenter. -->`

Chapitre 2 : Mise en forme du texte

1. Formatage de texte simple

Votre page contiendra très certainement du texte. Pour ne pas lasser le lecteur, ou pour mettre un mot en évidence, vous voudrez changer le style de votre texte.

Les balises suivantes vous seront utiles :

Balises Signification Résultats

- `Oui` Pour mettre un texte en caractère gras Oui
- `Non` Vous pouvez également utiliser la balise Non
- `<i>Vous</i>` Pour mettre un texte en italique Vous
- `Nous` Vous pouvez également utiliser la balise Nous
- `<u>hier</u>` Pour souligner un texte Hier

Pour barrer un texte (1 et 2)

- `hier` (1)
- `<srike>hier</strike>` (2)
- `1^{er}` Mettre un texte en exposant 1^{er}
- `H₂O` Mettre un texte en indice H₂O
- `<big>Grand</big>` Pour agrandir votre texte Grand
- `<small>Petit</small>` Pour rapetisser votre texte Petit
- `<cite>Citation</cite>` Pour signaler une citation Citation

Il n'est pas rare d'utiliser plusieurs balises pour un même élément de texte. Il faut veiller à bien respecter l'ordre des balises entrantes et sortantes.

`<i>Bon</i>`

Bon est correct.

`<i>...</i>` fonctionne mais est fortement déconseillé.

Les titres

Pour mettre en évidence vos données, les balises de titre sont les plus adaptées.

- `<h1>Titre 1</h1>`
- `<h2>Titre 2</h2>`
- `<h3>Titre 3</h3>`
- `<h4>Titre 4</h4>`
- `<h5>Titre 5</h5>`
- `<h6>Titre 6</h6>`

Il faut savoir qu'en les utilisant, un saut de paragraphe est généré par le navigateur après la balise sortante.

2. Quelques balises plus particulières

Les navigateurs ne reconnaissent qu'un espace entre les mots. Ce qui peut se révéler gênant dans certaines situations.

La balise `<pre>` `</pre>` affiche un texte dit pré-formaté. Le navigateur prend ainsi en compte tous les espaces et sauts de ligne définis à l'écran. La balise `<adress>...</adress>` vous permet d'indiquer une adresse.

Un saut de ligne se fait avec une des rares balises non fermantes `
`. Il ne faut pas confondre le saut de ligne avec la balise paragraphe `<p>`.

3. Mise en forme des caractères

Bien que bientôt reléguée aux oubliettes par les CSS, voici la balise qui vous permettra de travailler vos caractères : ``

Les attributs

- **Size** permet de définir la taille de la fonte à afficher. Elle peut être indiquée de plusieurs façons :
 - avec une taille prédéfinie allant de 1 à 7 , la valeur par défaut étant 3.
 - de façon relative par rapport à la valeur par défaut soit des chiffres allant de -3 à +3.

Exemples de définition de taille

```
<font size="1">Taille</font> donne Taille 1
<font size="2">Taille</font> donne Taille 2
<font size="3">Taille</font> donne Taille 3
<font size="4">Taille</font> donne Taille 4
<font size="5">Taille</font> donne Taille 5
<font size="6">Taille</font> donne Taille 6
<font size="7">Taille</font> donne Taille 7
```

Exemple de modification de taille

```
<font size="-3">Taille</font> donne Taille .3
<font size="-2">Taille</font> donne Taille .2
<font size="-1">Taille</font> donne Taille .1
```

Taille normale

```
<font size="+1">Taille</font> donne Taille +1
<font size="+2">Taille</font> donne Taille +2
<font size="+3">Taille</font> donne Taille +3
```

- **Face** permet de définir la fonte à utiliser pour l'affichage. Vous pouvez utiliser toutes les polices de caractères que votre système d'exploitation supporte mais vous devez savoir que celui qui visite votre page ne supportera peut être pas les polices que vous avez utilisées. Pour pallier à ce petit inconvénient, vous pouvez indiquer des polices de remplacement en les séparant par des virgules. Une bonne façon de faire est d'utiliser la police standard de plusieurs systèmes d'exploitation.

Voici les polices les plus courantes ainsi que leurs équivalents sur d'autres plateformes.

- Helvetica, sans-serif
- Courier New, Courier, mono
- Georgia, Times New Roman, Times, serif

- Times New Roman, Times, serif
- Verdana, Arial, Helvetica, sans-serif

Exemple

```
<font face="Arial, Helvetica, sans-serif">Arial</font> Arial
<font face="Courier New, Courier, mono">Courier New</font> Courier New
<font face="Georgia, Times New Roman, Times, serif">Georgia</font>
<font face="Times New Roman, Times, serif">Times New Roman</font>
<font face="Verdana, Arial, Helvetica, sans-serif">Verdana</font>
```

- **Color** permet de définir la couleur d'affichage.

Voici les codes de quelques couleurs :

- ✓ **Bleu #0000FF**
- ✓ **Blanc #FFFFFF**
- ✓ **Rouge #FF0000**
- ✓ **Gris clair #C0C0C0**
- ✓ **Vert #00FF00**
- ✓ **Violet #8000FF**
- ✓ **Jaune #FFFF00**
- ✓ **Noir #000000**

Exemple

```
<font color="#0000FF">Bleu</font> Bleu
<font color="#FF0000">Rouge</font> Rouge
<font color="#00FF00">Vert</font> Vert
<font color="#000000">Noir</font> Noir
```

4. Remarques importantes

Voici quelques raisons pour ne plus utiliser cette balise:

- a. Elle est considérée comme étant obsolète ou déconseillée d'emploi par le W3C. Les feuilles de style (CSS) la remplacent.
- b. Elle rend difficile la mise à jour d'un site car les balises FONT étant intégrées à chaque page, lorsqu'on souhaite modifier la taille ou la couleur d'un titre sur un site entier, il est nécessaire de modifier chaque page individuellement.
- c. Elle est interprétée différemment par les différentes plates-formes (notamment Mac et PC)

Les raisons pour éventuellement l'utiliser :

- a) Elle est compatible avec les versions anciennes version des navigateurs.
- b) Elle est plus facile à apprendre pour un débutant.

Chapitre 3 : La mise en page

1. Mise en forme des paragraphes

Les paragraphes de vos pages HTML peuvent être mis en forme de différentes façons. L'attribut `ALIGN="Type d'alignement"` se charge de cette opération. Il est normalement inclus dans la balise paragraphe (`<p>`). Il peut se présenter de deux manières :

`<p style="text-align: left">` ou `<p align="left">`

Mais peut aussi être inclus dans la balise `<div>` :

`<div style="text-align: left">`

ou

`<div align="left">`

2. Les listes

a) Les listes ordonnées

Une liste ordonnée est gérée par la balise `` qui permet d'avoir une liste numérotée qui s'incrémente automatiquement. La balise `` est définie avec `` qui représente un item (énumération). La balise `` indique le titre de la liste.

Exemple d'une liste ordonnée avec 3 items :

```
<ol>
<li>Titre
    <li>item 1
    <li>item 2
    <li>item 3
</ol>
```

Résultat:

Titre
1. item 1
2. item 2
3. item 3

Les autres attributs pour la balise `` :

`type=a` : Marqueurs en majuscules

`TYPE=a` : Marqueurs en minuscules

`TYPE=I` : Marqueurs en chiffres romains (majuscules)

`TYPE=i` : Marqueurs en chiffres romains (minuscules)

`TYPE=1` : Marqueurs en nombres (choix par défaut)

`START=n` : Valeur de départ du marqueur

Exemple d'une liste en chiffres romains (majuscules) démarrant au chiffre 2 :

```
<ol type=I start=2>
```



```
<li>item 1
```

```
<li>item 2
```

```
<li>item 3
```

```
</ol>
```

Résultat :

II. item 1

III. item 2

IV. item 3

b) Les listes non ordonnées

Une liste non ordonnée s'obtient avec la balise `` qui permet d'avoir une liste, précédée d'une puce. La balise `` est définie avec `` qui représente un item (énumération). La balise `<lh>` indique le titre de la liste.

Exemple d'une liste non ordonnée avec 3 items :

```
<ul>
```

```
<lh>Titre
```

```
<li>item 1
```

```
<li>item 2
```

```
<li>item 3
```

```
</ul>
```

Résultat :

· item 1

· item 2

· item 3

Les autres attributs pour la balise `` :

Compact : Spécifie une liste compacte

.type=disc : Puce en forme de disque plein

type=circle : Puce en forme de disque vide

type=square : Puce en forme de carré

Exemple d'une liste compacte, avec des puces en forme de disque vide :

```
<ul compact type=circle>
```

```
<li>item 1
```

```
<li>item 2
```

```
<li>item 3
```

```
</ul>
```

Résultat :

o item 1

o item 2

o item 3

c) Liste avec définition

Une liste de termes et de définitions correspondantes s'obtient avec la balise `<dl>` qui est accompagnée des balises `<dt>` et `<dd>`. `<dt>` représente le terme et `<dd>` la définition du terme. A noter que la balise `<dl>` n'a que l'attribut compact.

Exemple d'une liste de définition avec 1 terme et 3 définitions de 3 termes :

```
<dl>
  <dt>Terme
    <dd>def 1
    <dd>def 2
    <dd>def 3
</dl>
```

Résultat :

Terme
def 1
def 2
def 3

Combiner les types de listes Bien entendu, vous pouvez combiner plusieurs listes.

Exemple d'une liste combinant une liste ordonnée avec une liste de définition :

```
<ol type=I>
  <li>item 1
    <dl>
      <dt>Terme
      <dd>def 1
    </dl>
</ol>
```

Résultat :

I. item 1
Terme
def 1

3. Créer vos propres puces

Deux solutions s'offrent à vous pour créer vos propres puces :

- Soit vous passez par l'attribut `src="puce.gif"` si, par exemple, le fichier graphique de vos puces personnalisées se nomme `puces.gif`, mais cet attribut est spécifique à Internet Explorer.
- Soit vous remplacez l'attribut de la balise `` par la balise `` (voir chapitre suivant), en définissant votre image comme un simple fichier gif. Cette solution est la meilleure car elle est valable pour tous les navigateurs. Attention de faire un break (`
`) après chaque ligne.

Exemple d'une liste non ordonnée avec un fichier gif :

```
<ul>
   Personnalisez
<br>
```

vos puces !!

Chapitre 4 : Les tableaux

1. Présentation des tableaux

La balise <table> permet d'insérer un tableau et c'est l'une des balises les plus utilisées (elle est devenue officielle avec la version 3.2 du HTML).

En effet, les tableaux permettent une meilleure présentation des informations dans une page Web, nombreux sont les sites qui utilisent les tableaux pour obtenir une mise en page de type "journal".

2. Création des tableaux

La création des tableaux se passe autour de 4 balises :

- <table> : Définit au navigateur la création d'un tableau
- <th> : Définit l'en-tête de chaque colonne
- <tr> : Définit une ligne
- <td> : Définit une cellule

A noter que la balise <caption> qui affiche le titre du tableau possède 2 attributs :

- top : Titre placé au-dessus du tableau (par défaut)
- bottom : Titre placé en dessous du tableau

Bien entendu, toutes ces balises doivent être fermées pour indiquer la fin du tableau, d'une colonne ou d'une cellule.

Note : Si vous voulez définir une police de caractère pour le tableau, vous êtes obligé d'en définir pour chaque cellule même si elle est identique, exemple cidessous.

<td>Cellule 1</td>

Exemple d'un tableau avec son code source :

Légende du tableau

Colonne 1 Colonne 2

Cellule 1 Cellule 2

Cellule 3 Cellule 4

```
<table border=1>  
<caption>Légende du tableau</caption>  
<tr>  
<th>Colonne 1</th>  
<th>Colonne 2</th>  
</tr>  
<tr>  
<td>Cellule 1</td>  
<td>Cellule 2</td>  
</tr>
```

```

<tr>
<td>Cellule 3</td>
<td>Cellule 4</td>
</tr>
</table>

```

3. Aligner les éléments d'un tableau

L'alignement des données s'obtient en jonglant avec les attributs **align** pour l'alignement horizontal et **valign** pour l'alignement vertical. Ces attributs ont plusieurs valeurs :

- **align=left** : (par défaut) Les données sont alignées à gauche de la cellule (pour td) ou de toutes les cellules (pour th ou tr)
- **align=right** : Les données sont alignées à droite de la cellule (pour td) ou de toutes les cellules (pour th ou tr)
- **align=center** : Les données sont centrées à l'intérieur de la cellule (pour td) ou de toutes les cellules (pour th ou tr)
- **valign=middle** : (défaut) Les données sont centrées verticalement à l'intérieur de la cellule (pour td) ou de toutes les cellules (pour th ou tr)
- **valign=top** : Les données sont en haut de la cellule (pour td) ou de toutes les cellules (pour th ou tr)
- **valign=bottom** : Les données sont en bas de la cellule (pour td) ou de toutes les cellules (pour th ou tr)
- **valign=baseline** : Toutes les données des cellules sont alignées sur une ligne commune (uniquement pour th ou tr)

Exemple d'un tableau où les cellules sont alignées différemment :

Légende du tableau

Colonne 1 alignée à droite

Colonne 2 centrée

Cellule 1 alignée à gauche et en haut

Cellule 2 centrée et en bas

Cellule 3 basée sur la ligne

Cellule 4 basée sur la ligne

```

<table border=1 width=400>
<caption>Légende du tableau</caption>
<tr>
<th align=right>Colonne 1 alignée à droite</th>
<th align=center>Colonne 2 centrée</th>
</tr>
<tr>
<td align=left valign=top>Cellule 1 alignée à gauche et en haut</td>
<td align=center valign=bottom>Cellule 2 centrée et en bas</td>
</tr>

```

```

<tr valign=baseline>
<td>Cellule 3 basée sur la ligne</td>
<td>Cellule 4 basée sur la ligne</td>
</tr>
</table>

```

4. Fusionner les lignes et colonnes

Les lignes et colonnes peuvent être fusionnées, afin d'obtenir des cellules plus ou moins grandes et accueillir des données. Ce fusionnement est réalisé avec l'attribut **rowspan** et **colspan**.

- Pour fusionner 2 cellules adjacentes sur une ligne, **colspan** est requis (pour th ou td)
- Pour fusionner 2 cellules adjacentes sur une colonne, **ROWSPAN** est requis (pour th ou td)

Exemple d'un tableau avec des fusions :

Cellule 1 2 cellules fusionnées (sur la même ligne)

Cellule 2 Cellule 3

Cellule 4 Cellule 5

2 cellules fusionnées (sur la même colonne)

```

<table border=1>
<tr>
<td>Cellule 1</td>
<td colspan=2>2 cellules fusionnées (sur la même ligne)</td>
</tr>
<tr>
<td>Cellule 2</td>
<td>Cellule 3</td>
<td rowspan=2>2 cellules fusionnées (sur la même colonne)</td>
</tr>
<tr>
<td>Cellule 4</td>
<td>Cellule 5</td>
</tr>
</table>

```

5. Arrière plan, mise en page, espacement d'un tableau

En principe, la hauteur et la largeur d'une cellule s'adaptent automatiquement aux données (texte ou image). Néanmoins, vous pouvez définir une hauteur et une largeur pour un tableau ou une cellule ; l'espacement des cellules et la taille de la bordure peuvent être aussi paramétrés.

Attributs de mise en page et d'espacement (valable pour <table>, <td> ou <th>) :

- **width** : Spécifie la largeur du tableau ou d'une cellule, en pixels ou en pourcentage de la fenêtre du navigateur

- **height** : Spécifie la hauteur du tableau ou d'une cellule, en pixels ou en pourcentage de la fenêtre du navigateur
- **border** : Spécifie la taille en pixels (par défaut 1) de la bordure du tableau
- **cellpadding** : Spécifie l'espace en pixels entre la bordure et le contenu de la cellule du tableau
- **cellspacing** : Spécifie l'espace en pixels entre les cellules du tableau
- **bgcolor** : Spécifie la couleur d'une cellule
- **bordercolor** : Spécifie la couleur de la bordure du tableau
- **bordercolorlight** : Spécifie la couleur aux points culminants de la bordure du tableau (effet 3d)
- **bordercolordark** : Spécifie la couleur de l'ombre de la bordure du tableau (effet 3d)
- **background** : Spécifie l'URL de l'image à afficher comme image d'arrière plan du tableau

I. Explorer seulement)

A noter que pour la balise <td> l'attribut **nowrap** permet d'éviter les retours à la ligne (inutile si vous avez défini votre tableau en pixels).

Exemple d'un tableau avec certains de ces attributs :

Cellule 1 Cellule 2 Cellule 3

Cellule 4 Cellule 5 Cellule 6

Cellule 7 Cellule 8 Cellule 9

```
<table border=2 bordercolor="blue" bordercolordark="aqua" width=300
height=150>
<tr>
<td bgcolor="pink">Cellule 1</td>
<td>Cellule 2</td>
<td>Cellule 3</td>
</tr>
<tr>
<td>Cellule 4</td>
<td bgcolor="yellow">Cellule 5</td>
<td>Cellule 6</td>
</tr>
<tr>
<td>Cellule 7</td>
<td>Cellule 8</td>
<td bgcolor="maroon">Cellule 9</td>
</tr>
</table>
```

6. Les attributs rules et frame (HTML 4.0)

Ces attributs sont valables uniquement pour la balise <table> et sont reconnus par les navigateurs de dernière génération qui savent exploiter le HTML 4.0 .

L'attribut frame :

Il fonctionne seulement si border est présent dans la balise <table> et signale les côtés d'un cadre à afficher. Voici la liste des valeurs qu'il peut supporter :

- **above** : Affiche les bordures externes du haut du tableau seulement
- **below** : Affiche les bordures externes du bas du tableau seulement
- **border** : Affiche les bordures sur tous les côtés du tableau
- **box** : Affiche une boîte autour du tableau
- **insides** : Affiche les bordures sur le dessus et le dessous du tableau seulement
- **hsides** : Affiche les bordures externes du côté horizontal du tableau (haut et bas)
- **lhs** : Affiche les bordures externes du côté gauche du tableau seulement
- **rhs** : Affiche les bordures externes du côté droit du tableau seulement
- **void** : Permet de retirer toutes les bordures externes du tableau
- **vsides** : Affiche les bordures externes des côtés droit et gauche du tableau

L'attribut rules :

Il fonctionne seulement si l'attribut border est présent dans la balise <table> . Il permet de spécifier des règles supplémentaires à l'intérieur du tableau. Les valeurs acceptées sont les suivantes :

- **none** : Pour retirer toutes les bordures internes du tableau
- **groups** : Affiche une bordure horizontale entre les sections
- **rows** : Affiche une bordure horizontale entre toutes les lignes du tableau
- **cols** : Affiche une bordure horizontale entre toutes les colonnes du tableau
- **all** : Affiche des bordures dans le tableau

A noter que ces deux attributs doivent aussi être utilisés avec les balises :

- **thead** : Définissant l'en-tête du tableau
- **tbody** : Définissant le corps du tableau
- **tfoot** : Définissant le pied de page du tableau

Exemple d'un tableau utilisant ces attributs :

Ligne 1 Cellule 1 Cellule 2

Ligne 2 Cellule 3 Cellule 4

```
<table border=2 width=200 frame=hsides rules=cols>
<thead>
<colgroup span=3 align=left valign=top>
</thead>
<tbody>
<tr>
<td>Ligne 1 </td>
<td>Cellule 1</td>
<td>Cellule 2</td>
</tr>
<tr>
<td>Ligne 2 </td>
<td>Cellule 3</td>
```

```
<td>Cellule 4</td>
</tr>
</tbody></table>
```

Chapitre 5 : Les images et maps

1. Les images

La balise pour introduire une image dans une page HTML est ****. C'est une balise ponctuelle. Ses principaux attributs sont :

- **src** qui détermine le chemin du fichier image. Ce chemin peut être indiqué en partant du répertoire du document, il se présentera alors sous la forme "../..images/fichier.gif" par exemple. Il peut également être indiqué en partant de la racine principale du site. Il aura alors la forme : "/exemple/images/fichier.gif" par exemple. Il commencera toujours par / dans ce cas.

La balise minimale pour une image se présente ainsi :

```

```

On obtient par exemple le résultat suivant:

- **height, width**

Lorsque vous introduisez plusieurs images sur votre page, vous verrez que lors de l'affichage, elles se mettent à danser. C'est très désagréable. Cela vient du fait que la taille de l'image n'est pas définie. En plus, cela ralentit l'affichage de la page.

height et **width** indiquent au navigateur la hauteur et la largeur de l'image (en pixel) qui va être affichée.

Exemple :

```

```

Pour ajuster une image à une dimension précise tout en conservant les proportions, il suffit de préciser soit la hauteur, soit la largeur et automatiquement l'une ou l'autre des dimensions s'alignera sur celle que vous aurez spécifiée.

Exemple :

```

```

- **alt** : Cet attribut permet l'affichage dans une bulle de texte. Lorsque l'affichage de l'image ne se déroule pas correctement, ce même texte apparaîtra à la place de l'image. Comme le net devient très fréquenté par les non-voyants, il est important d'utiliser cet attribut pour expliquer l'image. Passez la souris sur la première image pour voir s'afficher la bulle. L'image du second exemple est volontairement inexistante pour montrer la manière dont cet attribut s'affiche.

Exemples:

```

```

```

```

- **longdesc** : Cet attribut définit l'adresse d'un document décrivant l' image.

Exemple : ``

- **lowsrc** : Cet attribut définit l'adresse d'une image à basse résolution.

``

- **name** : Cet attribut permet de un nom à votre image. Elle est surtout employée lorsque vous voulez travailler avec du JavaScript.

``

- **border** : Cet attribut permet de définir (en pixel) une bordure à votre image.

``

Elle est assez peu employée sous cette forme. Lorsque vous utilisez une image comme lien, le navigateur affiche automatiquement une bordure.

` `

La définition d'une bordure à 0 pixels permet d'éviter ce cadre disgracieux.

` `

- **align** : La position de l'image dans votre page est déterminée par cet attribut. Deux positions sont fréquemment utilisées seule : gauche ou droite.

``

- **text** : Cet attribut prend vraiment toute son importance lorsque l'image est accompagnée de texte.

Exemples d'alignement :

- Par défaut : `align="baseline"` - En haut : `align="top"` - Au milieu : `align="middle"`

- **hspace, vspace**

Ces deux attributs permettent de définir (en pixels) une marge entre l'image et le reste du contenu. La lisibilité du document en est fortement améliorée.

Exemple :

`<p align="justify">`

``

Hyper Text Markup Language ou langage hypertexte est le langage à partir duquel sont créées les pages du web. Un site web est constitué d'un ou de plusieurs documents HTML, appelées aussi pages. Pour se déplacer d'une page à une autre on passe par l'intermédiaire d'hyperliens. Un document HTML est un document qui a l'extension HTML ou HTM indifféremment. C'est un simple fichier texte avec l'extension htm ou html.

`</p>`

Exemple : Les images et Internet Explorer Avec Internet Explorer 6, est apparue la barre d'images. Si vous désirez la désactiver de certaines images tout en la laissant sur d'autres, rajoutez simplement `galleryimg="no"` ou `galleryimg="false"` dans la balise de l'image.

Promenez votre souris sur les différentes images générées par le code ci-dessous pour voir le résultat.

```

```

```

```

2. Les maps

La balise permet de créer une image réactive (côté client) en HTML. L'image réactive est utilisée généralement pour définir plusieurs liens (interne ou externe) sur une même image. Elle est souvent utilisée pour des banderoles. La balise **<map>** doit être utilisée avec la balise **<area>** qui, elle, possède trois attributs :

- **Shape** : qui peut prendre trois valeurs :

 - rect** : Pour un rectangle

 - circle** : Pour un cercle

 - poly** : Pour un polygone

- **coords** : Indique les coordonnées :

Des coins supérieurs gauche et inférieur droit de la zone pour un rectangle rect, de chaque angle du polygone si vous optez pour la valeur poly, du centre du cercle et la dimension du rayon de ce même cercle pour la valeur circle.

- **href** : définit la page de destination du lien correspondant à la zone. Vous pouvez bien entendu, utiliser un lien interne ou externe.

Exemple d'une image réactive, composée d'un rectangle (zone 1) et d'un cercle (zone 2).

La zone 1 revient au début de la page et la zone 2 se retrouve à nouveau au niveau du titre "La balise **<map>** et ses attributs".

Voici le code de l'exemple ci-dessus :

```
<map name="testmap">
```

```
<area shape="rect" coords="10,9,58,27" href="#top">
```

```
<area shape="circle" coords="104,18,14" href="#map">
```

```
</map>
```

```

```

Note : on remarque que l'attribut name (à l'intérieur de la balise map) et l'attribut usemap (à l'intérieur de la balise img) ont le même nom, ceci afin de les relier (obligatoire).

Chapitre 6 : Les liens

1. La balise <a> et ses attributs

La balise <a> permet d'établir un lien hypertexte (l'essence même du Web), à l'intérieur même de votre document vers un autre document, ou encore vers l'extérieur (vers un autre site Web).

Ses attributs principaux sont :

- **href** : Permet d'indiquer la source du document (relative ou absolue)
- **target** : Permet de définir la source d'un cadre comme document (dans la cas d'une frame)
- **style** : Permet de spécifier une feuille de style
- **title** : Permet de spécifier une description du lien (identique à l'attribut ALT pour une image et uniquement avec Internet Explorer)

2. Les liens hypertextes

Les liens internes (même page)

Exemple :

```
<a href="page.html">Page</a>
```

Les liens internes permettent de se déplacer à l'intérieur d'un fichier HTML, sans que le visiteur soit obligé de faire "scroller" la fenêtre. Le visiteur va donc cliquer sur un lien et il sera amené vers l'ancre. Pour cela deux étapes sont nécessaires : Définir la cible d'un lien (l'ancre) et associer ce même lien à cette ancre.

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

Exemple :

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

Il n'est pas nécessaire, dans ce cas précis, de mettre un texte ou une image, entre les balises d'ouverture et de fermeture (le nom ne doit pas contenir ni d'espaces, ni d'accents, ni de caractères spéciaux).

- Associer un lien à un ancre

Exemple :

```
<a href="#top">top</a>
```

Les liens internes (pages différentes)

- Lien vers une page située dans le même répertoire

Exemple :

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

- Lien dans un répertoire différent

Exemple :

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

- Lien dans un même répertoire et vers une ancre d'une autre page

Exemple :

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

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

Exemple :

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

- Les liens externes

Les liens externes commencent comme les autres liens, c'est à dire avec la balise `` . Par contre suivant le protocole, le reste de la syntaxe varie.

- Lien externe vers un site http

Exemple :

```
<a href="http://www.yahoo.fr">Découvrez YAHOO France</a>
```

- Lien externe vers un serveur FTP

Exemple :

```
<a href="ftp://ftp.allhtml.com">FTP ALL HTML</a>
```

- Lien externe vers un serveur de news (Attention pour que ce lien soit valide il faut que le visiteur soit abonné à ce serveur de news)

Exemple :

```
<a href="news:news.site">Serveur de news</a>
```

Note : Avec un lien externe vous pouvez également proposer un fichier (.exe ou .zip) en téléchargement

Exemple :

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

- Les liens e-mail

Vous pouvez créer un lien e-mail pour que la messagerie paramétrée par défaut du visiteur se lance automatiquement. Attention tout de même : ce lien ne marche pas avec les premières versions de Netscape et d'I.Explorer. Il possède plusieurs attributs dont : **?subject=** pour définir un sujet (ou objet) automatiquement, ce qui est très pratique pour trier votre courrier par le biais de votre logiciel classique.

- Lien e-mail sans sujet

Exemple :

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

- Lien e-mail avec sujet (ici "renseignement")

Exemple :

```
<a href="mailto:nom@provider.com?subject=renseignement">nom@provider.com</a>
```

- Lien e-mail avec sujet, et texte dans le corps du message

Exemple :

```
<a href="mailto:nom@provider.com?subject=renseignement&body=Votre message">nom@provider.com</a>
```

- Lien e-mail avec sujet, envoyé à 2 personnes (dont une en CC)

Exemple :

```
<a href="mailto:nom@provider.com?cc=nom2@provider2.com&subject=renseignement">
nom@provider.com</a>
```

- Lien e-mail avec 2 personnes (dont une en CCI ou BCC)

Exemple :

```
<ahref="mailto:nom@provider.com?bcc=nom2@provider2.com&subject=renseignement" >
nom@provider.com</a>
```

Chapitre 7 : Les formulaires

1. Rôles

Les formulaires permettent avant tout de faire réagir votre visiteur, de l'inviter à laisser des informations (données, impressions, commentaires etc.). La création des formulaires se réalise grâce à plusieurs champs, dans lesquels le visiteur entre une information ou opte pour un choix. L'information est ensuite envoyée, à l'aide de scripts, sur le serveur qui vous héberge puis renvoyée, dans la plupart des cas, sur l'adresse électronique de votre choix.

Un formulaire peut servir à recueillir les avis des visiteurs, leur e-mail en vue d'une newsletter. Mais ça peut être aussi un bon de commande, un sondage, un jeu... Un formulaire peut aussi être une passerelle entre le visiteur et des applications écrites en PHP, ASP, Java ou JavaScript.

La balise principale <FORM>

Le formulaire est délimité dans le code HTML par le conteneur : <FORM>

</FORM>, c'est-à-dire que toutes les balises de base propres aux formulaires devront être contenues entre ce couple de balises. A l'intérieur de ce conteneur pourra aussi figurer les autres balises HTML pour la mise en forme (fontes, couleurs, tableaux, images, liens...).

Les principaux attributs de <FORM>

La balise <FORM> possède des attributs dont-on citera quelques-uns :

- **Name** : attribut un nom au formulaire.
- **Action** : spécifie l'adresse (URL ou e-mail) vers laquelle les données seront envoyées. Adresse le formulaire vers l'URL (relative) du script (fourni par le serveur qui héberge vos pages), qui accepte l'information et vous le retourne suivant les paramètres définis. Les valeurs les plus couramment utilisées sont :

☒ "mailto:webmaster@seneweb.sn" : Pour l'envoi du formulaire par le logiciel de messagerie de votre visiteur.

Attention : cette valeur peut poser problème avec certains navigateurs,

☒ "/cgi-bin/mailer" : Pour l'envoi du formulaire par le script CGI de votre hébergeur.

- **Method** : sélectionne une méthode d'envoi des données. Deux valeurs sont proposées :

- ✓ post : Pour un formulaire classique
- ✓ get : Pour (par exemple) définir un moteur de recherche sur votre site
- **Enctype**: Cet attribut spécifie le format des données envoyées, dans le cas où un protocole n'imposerait pas de format précis. Il possède plusieurs valeurs dont :
 - ✓ "text/plain" : Valeur utilisée dans le cas d'un formulaire classique
 - ✓ "multipart/form-data" : Valeur utilisée dans le cas où le formulaire comprendrait un fichier attaché (<input type="file" ...>)

Exemple pour le serveur d'hébergement Multimania :

```
<form method=get action="/cgi-bin/mailer">
```

Exemple pour un formulaire classique sans l'aide de script CGI :

```
<form action="mailto:webmaster@seneweb.sn" method="post" enctype="text/plain">
```

2. Les champs de saisie

La balise `<input type="text" ... >` permet à l'utilisateur de saisir une ligne de texte. Elle a les attributs suivants :

- name : Définit le nom de la donnée (obligatoire)
- size : Définit la taille visible du champ en nombre de caractères
- maxlength : Taille maximale de caractères autorisés
- value : Définit le texte par défaut à l'intérieur du champ

A noter que la valeur type peut avoir plusieurs valeurs dont :

- password : Code secret
- range : Permet de définir un espace de saisie numérique (exemple entre 0 et 50, min="0" max="50")

Exemple d'un champ de saisie et son code source :

```
http://<input type="text" name="urlsite" size=40 maxlength=40 value="http://">
```

3. Boîte de texte multiligne

C'est la balise `<textarea>` qui est utilisée pour réaliser une boîte de texte multiligne. Cette balise est surtout choisie pour permettre aux visiteurs de votre site de laisser un commentaire ou de taper un texte quelconque.

Ses attributs sont :

- name : Définit le nom de la donnée (obligatoire)
- rows : Définit le nombre de lignes dans la boîte de saisie
- cols : Définit le nombre de colonnes (en caractères) dans la boîte de saisie
- readonly : Spécifie que la boîte multiligne est en mode lecture seulement.

A noter que si du texte se trouve à l'intérieur des balises `<textarea>` et `</textarea>` il va alors apparaître comme texte par défaut.

Exemple d'une boîte de saisie et son code source :

Tapez votre commentaire `<textarea name="commentaire" cols=30 rows=4>`

Tapez votre commentaire

`</textarea>`

4. Listes de choix

La balise **<select>** présente une liste de propositions, soit sous forme de liste déroulante (un seul choix possible), soit sous forme de menu (plusieurs choix possibles), elle doit être couplée avec la balise **<option>** pour définir les choix.

Voici ses attributs :

- **name** : Définit le nom de la donnée (obligatoire)
- **size** : S'il a comme valeur 1(valeur par défaut), c'est une liste déroulante ; si size supérieur à 1 c'est une liste de choix avec une barre de défilement
- **multiple** : Ce choix est valable uniquement si size est supérieur à 1

La balise **<option>** comporte deux attributs :

- **value** : Nom de la donnée pour le choix
- **selected** : S'il est présent dans la balise **<option>**, le choix est alors reconnu comme choix par défaut

Exemple d'une liste déroulante et son code source :

Choix 1

```
<select name="liste">
  <option value="Choix 1">Choix 1
  <option value="Choix 2">Choix 2
  <option value="Choix 3">Choix 3
</select>
```

Exemple d'une liste sous forme de menu et son code source :

Choix 1

Choix 2

Choix 3

```
<select name="liste" size=3 multiple>
  <option value="Choix 1">Choix 1
  <option value="Choix 2">Choix 2
  <option value="Choix 3">Choix 3
</select>
```

5. Cases à cocher et boutons radio

La Balise **<input>**, que l'on a vu au -dessus, permet aussi de représenter une liste de cases à cocher (plusieurs choix possibles) ou une liste dite "bouton radio" (un seul choix possible). Il suffit pour cela de lui spécifier l'attribut **type="checkbox"** (cases à cocher) ou **type="radio"** (bouton radio).

Les autres attributs sont les suivants :

- **name** : Définit le nom de la donnée (obligatoire)
- **value** : Nom de la donnée pour le choix
- **checked** : S'il est présent dans la balise **<input>**, le bouton sera coché par défaut

Exemple d'une liste de cases à cocher et son code source :

Cette liste se positionne verticalement du fait que la balise **
** a été insérée après chaque choix.

CASE 1

CASE 2

CASE 3

```
<input type="checkbox" name="CHOIX" value="CASE 1"> CASE 1<br>
<input type="checkbox" name="CHOIX" value="CASE 2"> CASE 2<br>
<input type="checkbox" name="CHOIX" value="CASE 3"> CASE 3<br>
```

Exemple d'une liste à boutons radio et son code source : A noter que le choix 3 est coché par défaut.

Bouton 1 Bouton 2 Bouton 3

```
<input type="radio" name="CHOX" value="Bouton 1"> Bouton 1
<input type="radio" name="CHOX" value="Bouton 2"> Bouton 2
<input type="radio" name="CHOX" value="Bouton 3" checked> Bouton 3
```

6. Fichier attaché

Lors d'un envoi de formulaire, vous pouvez proposer au visiteur de vous faire parvenir un fichier attaché (ou pièce jointe), il suffit pour cela d'indiquer la valeur "file" à l'attribut type.

Exemple pour un choix de fichier attaché :

```
<input type="file" value="fichier">
```

7. Soumettre un formulaire

La soumission (envoyer) ou la remise à zéro (recommencer) d'un formulaire se fait, une fois de plus, avec la balise <input> type, ayant ici deux autres valeurs :

- **submit** : Envoi le formulaire vers le serveur
- **reset** : Remet à zéro les choix du formulaire

Les légendes inscrites sur les boutons peuvent être modifiées avec l'attribut value.

Exemple des 2 boutons et leur code source :

Envoyer Recommencer

```
<input type="submit" value="Envoyer">
<input type="reset" value="Recommencer">
```

Personnalisation de boutons

Vous pouvez créer votre propre bouton d'envoi et de remise à zéro, ce qui mettra une touche personnelle à votre formulaire.

- ⊕ Pour obtenir un bouton personnalisé de remise à zéro (**reset**), il suffit tout simplement d'encadrer l'image de votre bouton (avec la balise ex :) entre les balises et .
- Cela aura pour effet de recharger votre page html de formulaire et donc d'annuler tous les choix du visiteur.

Exemple d'un bouton "reset" personnalisé et son code source :

```
<a href="formul2.htm"> </a>
```

- ⊕ En ce qui concerne le bouton d'envoi, il faut une fois de plus passer par la balise

input avec l'attribut type="image".

Exemple d'un bouton d'envoi personnalisé et son code source :

```
<input type="image" src="gif/envoyer.gif" border=0 width=45 height=9  
alt="Envoyer">
```

8. Inclure des données cachées

Si vous souhaitez envoyer certaines données au script du serveur chargé de traiter votre formulaire, vous devez utiliser la balise **<input>** avec l'attribut **type="hidden"**. Prenez contact avec le service qui vous héberge afin d'obtenir les différentes valeurs à indiquer avec la balise **<input>**.

9. Mise en forme des formulaires

Il y a mille et une façons de concevoir un formulaire. Mais sachez, avant tout, que plus il sera clair et lisible, plus il sera complété par vos visiteurs. Pour cela, vous pouvez opter pour des tableaux, des listes à puces (pour les cases à cocher notamment), des listes de définition et bien sûr, des balises de formatage et d'alignement de texte.

Exemple de formulaire :

Voici un exemple concret de formulaire ainsi que le code source :

```
<form action="mailto:email@adresse.com" method=post enctype="text/plain">  
Votre E-Mail. <input type="text" name="email" size=30 maxlength=40  
value="nom@provider.domaine">  
<BR><br>  
Tapez ici votre commentaire. <textarea name="comm" cols=31 rows=4> Votre  
commentaire </textarea> <br> <br>  
<input type="submit" value="Envoyer">  
<input type="reset" value="Annuler">  
</form>
```