

Chapitre I : Langage HTML 4

1. Introduction

Le mot HTML vient de l'anglais : HyperText Markup Language. Le HTML n'est pas un langage de programmation mais c'est un langage standard qui permet de définir l'habillage d'un document (page Web). C'est à dire : la façon dont il doit s'afficher à l'écran d'un navigateur.

Ainsi, une page HTML est un fichier texte enrichi de codes et de commandes appelées balises. Le fichier source porte généralement l'extension : .htm ou .html

Même s'il existe des éditeurs WYSIWYG : What You See Is What You Get (Ce que vous voyez est ce que vous obtenez), on peut écrire du code HTML à l'aide de n'importe quel éditeur de texte.

Le code source doit être sauvegardé sur le serveur Web. Il est envoyé vers les clients à l'aide du protocole HTTP. Par la suite, le code HTML est interprété par le navigateur et l'affiche sur écran sous forme de : textes mis en forme, images, animations, multimédias, ...

Le code source est visible aux utilisateurs mais ils ne pourront pas le modifier (facilement) sur le serveur.

2. Notion de Balise

La balise en HTML est un mot clé du langage qui est encadré par les symboles < et >. Pour la plupart des balises, il existe une balise de début et une balise de fermeture précédée par / : comme par exemple <html> ... </html>. Par contre, certaines balises n'ont pas de balises fermantes : comme par exemple ,
, ... Pour d'autres, la balise fermante est facultative : comme par exemple <p>, <td> ...

Il est possible d'imbriquer les balises de la manière suivante : <balise1><balise2> ... <balisen> ... </balisen> ... </balise2></balise1>. La fermeture se fait dans l'ordre LIFO (Pile). Par contre, l'imbrication qui consiste à insérer une balise à l'intérieur d'une autre est interdite. Par exemple, ceci est faux : <balise1 ... <balise2> ... >.

2.1 Attribut de balise

Pour la majorité des balises, il existe un ou plusieurs attributs qui permettent de définir certaines propriétés supplémentaires. Généralement, les attributs sont facultatifs mais parfois ils sont obligatoires.

Les attributs sont des mots réservés. A chaque attribut, on affecte une valeur comme suit:

<Nom_Balise att_1="val_1" att_2="val_2" ... att_n="val_n">

Certains attributs (cas rares) n'ont pas de valeurs : comme par exemple <hr noshade>.

Chaque balise a ses propres attributs. Certains attributs sont communs entre plusieurs balises : **color**, **align**, ... On peut trouver aussi des balises qui n'admettent pas d'attributs.

2.2 Valeurs d'attributs

Certaines valeurs doivent appartenir à un ensemble fini : Comme dans le cas des valeurs de l'attribut **align** sont dans l'ensemble : {"left", "right", "center"}. Certaines valeurs doivent obéir à une syntaxe bien précise : Comme pour l'attribut **color** = "#FF0080". Par contre, d'autres valeurs sont libres : **name** = "info acad".

Ci-dessous seront expliqués quelques valeurs particulières utilisées souvent en HTML.

2.2.1 Les valeurs « couleurs »

On peut donner la valeur d'une couleur de deux manières différentes : par nom ou par code RGB.

A travers le nom de cette couleur comme : **white black blue green red**...

Par code RGB, il faut respecter la syntaxe suivante : **#RRGGBB**. RR, GG ou BB est une valeur hexadécimale entre 00 (correspond à 0) et FF (255 décimal).

- RR est le dosage de la couleur rouge,
- GG est le dosage de la couleur verte,
- BB est le dosage de la couleur bleue.

Le tableau suivant donne la liste des noms des couleurs de la palette Web (16 couleurs) :

La couleur	Nom HTML	Code RGB
Blanc	White	#FFFFFF
Argent	Silver	#C0C0C0
Gris	Gray	#808080
Noir	Black	#000000
Rouge	Red	#FF0000
Marron	Maroon	#800000
Jaune	Yellow	#FFFF00
Vert citron	Lime	#00FF00
Vert	Green	#008000
Olive	Olive	#808000
Cyan	Aqua	#00FFFF
Bleu gris	Teal	#008080
Bleu	Blue	#0000FF
Bleu marine	Navy	#000080
Magenta	Fuchsia	#FF00FF
Pourpre	Purple	#800080

2.2.2 Valeurs « Noms de fichiers »

On peut donner la valeur d'un fichier de deux manières différentes : en relatif ou en absolu.

Pour la première façon, le fichier doit être disponible sur le même serveur Web que le fichier contenant le code HTML. Si le fichier est sur le même répertoire (dossier), on donne directement le nom du fichier : Par exemple "**image.jpg**". Dans le cas où le fichier serait sur un répertoire fils, on doit noter le nom de ce répertoire suivi du symbole / pour descendre dans l'arborescence : Par exemple "**images/image.gif**". Mais si le fichier est sur un dossier parent, on utilise **../** pour remonter dans l'arborescence : Par exemple "**../page.html**". Comme, on peut combiner les deux : par exemple "**../dossier/page.htm**".

Pour la seconde façon (absolu), elle est utilisée si le fichier est sur un autre serveur Web. Dans ce cas, on donne l'adresse complète du fichier : Par exemple "**http://www.nomsite.org/doss/.../fich.ext**".

Remarque : Il faut faire très attention à l'écriture et surtout à la casse (majuscules et minuscules).

2.2.3 Valeurs « taille »

On peut donner la valeur de la taille d'un élément de deux manières différentes : en absolu ou en relatif.

Pour donner une taille en **absolu**, on donne une valeur entière positive ou nulle *n* et ça correspond à une taille de *n* pixels. Pour donner une taille en **relatif**, on donne une valeur entière positive ou nulle suivi du symbole % (*n*%) et ça correspond à *n*% de l'espace disponible.

2.2.4 Valeurs « alignement »

Pour l'alignement horizontal d'un élément, les valeurs suivantes peuvent être utilisées : "**left**" : à gauche, "**right**" : à droite, "**center**" : centré.

Pour l'alignement vertical d'un élément, les valeurs suivantes peuvent être utilisées : "**top**" : en haut, "**middle**" : au milieu, "**bottom**" : en bas.

3. Structuration d'une Page HTML

Une page HTML bien structurée doit obéir à la syntaxe suivante :

```
<html>
  <head>    <!-- en-tête -->
    <title>  Titre de la page </title>
  </head>
  <body>    <!-- Corps de la page -->
    ...
  </body>
</html>
```

Les parties essentielles d'une page HTML sont : l'en-tête et le corps.

Remarques :

- L'indentation facilite la lecture du code.
- On peut écrire les balises en majuscules ou en minuscules. (L'interpréteur du HTML ne tient pas compte de la casse)
- Les images, animations, sons, vidéos, ... sont insérés à partir de fichiers externes et ne font pas partie du code source HTML.

3.1 L'en-tête

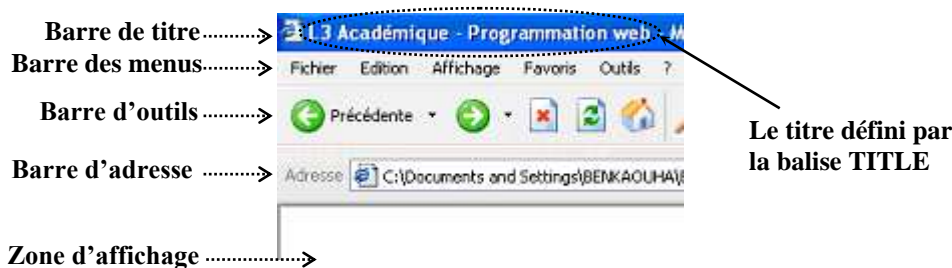
Le début de l'en-tête est défini par la balise `<head>` et sa fin est marquée par la balise `</head>`.

La majorité des éléments de l'entête ne sont pas affichés par le navigateur. Certains éléments sont liés à la page, certains sont des informations destinées aux moteurs de recherches, d'autres sont utilisés pour réaliser des traitements sur la machine cliente (petites animations ou défilements, vérifications, ...)

Les éléments de l'en-tête peuvent être classés en trois grandes classes : le titre, les metas et les scripts.

3.1.1 Le titre de la page

Il apparaîtra sur la barre de titre du navigateur. On l'insère entre les balises `<title>` et `</title>`. Par exemple, insérer dans l'en-tête d'une page Web : `<title> L3 Académique - Programmation Web</title>` donne l'affichage suivant :



3.1.2 Fond sonore

On peut aussi insérer un son de fond dans la page à l'aide de la balise suivante : `<bgsound src="fichier son" loop="nb_repetitions">`. Si on met loop à -1 le son se répète de façon infinie. La valeur de `src` obéit à la syntaxe des fichiers.

3.1.3 Script coté client

On peut insérer un script coté client (en JavaScript par exemple) entre les balises `<script>` et `</script>`. Nous reviendrons sur l'insertion des scripts dans le chapitre « JavaScript »

3.1.4 Les META

Les balises `<meta>` permettent d'insérer des informations concernant le document. Elles permettent aussi de noter diverses instructions utiles pour les navigateurs et pour les programmes d'indexation automatique utilisés par les moteurs de recherche (Robots). Il n'existe pas de standard pour les balises `<meta>`. C'est pour cette raison que nous citerons juste les principales utilisations.

a - Pour les moteurs de recherche

La forme générale est la suivante : `<meta name="nom_meta" content="valeur">`.

L'attribut `content` dépendra de la valeur de l'attribut `name` selon le tableau ci-dessous :

Name	Content
"description"	Un texte descriptif du contenu de la page Web. Cette description apparaîtra dans le résultat pour plusieurs moteurs de recherche.
"keywords"	Une suite de mots clés séparés par une virgule ou point virgule. Un utilisateur qui tape un de ces mots clés dans un moteur de recherche, cette page a de grande chance d'être trouvée. Un mot clé peut être un mot simple (taille) ou un mot composé (en-tête ou grand père).
"date"	Doit être de la forme : AAAA-MM-JJTHH:MM:SS+HH:MM Où AAAA est l'année sur 4 positions, MM le mois sur 2 position (00 représente <i>Janvier</i> et 11 pour <i>Décembre</i>), HH:MM:SS l'heure (en heures, minutes et secondes), après le signe + on donne le décalage par rapport à l'heure GMT. Il est possible de ne mentionner que la date sans l'heure, il suffit de la noter sous la forme AAAA-MM-JJ .
"author"	Le nom de l'auteur (c'est à dire: du responsable du contenu).
"reply-to"	L'adresse électronique (email) de contact.
"robots"	"all" : Le robot indexe tout (par défaut). "index" : Autoriser l'indexation de la page. "none" : Pour empêcher les moteurs de recherche d'indexer (rajouter à la base de données) la page actuelle. Ainsi, la page n'est pas affichée dans les résultats de recherche. "noindex" : Equivalente à "none" seulement permet de suivre les liens de la page. "nofollow" : Autorise le robot du moteur de recherche de transmettre le contenu du fichier HTML actuel à sa base de données mais l'empêche cependant de visiter les fichiers référencés par des liens dans la page actuelle. "follow" : Permet l'indexation de la page actuelle et de suivre ses liens.
"revisit-after"	"n days" où <i>n</i> est un entier positif. Pour indiquer aux moteurs de recherche de revenir dans <i>n</i> jours pour relire le fichier complètement.
"copyRight"	Les informations concernant la protection du contenu (Copyright).
"generator"	Le nom de l'application qui a généré cette page.
"progId"	L'identificateur du programme qui a généré cette page.
"title"	Le titre de la page. Ceci n'est pas une redondance avec la balise <title> car il peut être utile pour donner la traduction du titre dans d'autres langues.

Pour la description, les mots clés ou le titre, il est possible de noter plusieurs meta avec des langues différentes. Il suffit de rajouter l'attribut **lang="code_langue"**. Pour la valeur **"code_langue"** des codifications à deux lettres ou à quatre lettres sont autorisées. Pour la première, une abréviation du nom de la langue à deux lettres : par exemple **en** pour l'anglais, **ar** pour l'arabe, **es** pour l'espagnol, **de** pour l'allemand, **fr** pour le français ou **it** pour l'italien. Pour la seconde, il est rajouté au code de la langue, une abréviation en deux lettres du pays : par exemple **en-us** pour l'anglais américain, **ar-dz** pour l'arabe algérien.

b – Pour les serveurs proxy

Les pages Web appelées souvent sont sauvegardées temporairement sur ce qu'on appelle des serveurs Proxy, on parle alors de cache Proxy. Il est possible d'indiquer aux serveurs proxy de ne pas garder de copie de la page en rajoutant la meta suivante : **<meta http-equiv="pragma" content="no-cache">**.

c - Pour les navigateurs

Il est possible de préciser quel jeu de caractères sera utilisé lors de l'affichage par le navigateur de la page. Il est aussi possible de donner les langages complémentaires par défaut au HTML comme pour les feuilles de style et les Scripts. Ceci permettra au navigateur d'interpréter correctement en cas de doute le contenu du fichier HTML.

La forme générale est la suivante : **<meta http-equiv="nom_meta" content="valeur">**. Le tableau suivant donne plus de détails :

http-equiv	Content
"content-type"	La valeur doit respecter la syntaxe suivante : "type_mime; charset=code_jeu_caracteres". Puisqu'il s'agit de page HTML, le type MIME est text/html. Par exemple : "text/html; charset=iso-8859-1" ou "text/html; charset=windows-1252"
"content-language"	Langue du contenu de la page. Le code de la langue est donné en abréviation de 2 lettres ou 4 lettres comme ça été expliqué ci-dessus pour l'attribut lang.
"Content-Script-Type"	La valeur doit être le type MIME du langage de script utilisé, par exemple : "text/javascript".
"Content-Style-Type"	La valeur doit être le type MIME du langage de styles utilisé, par exemple : "text/css".

Les navigateurs sauvegardent les pages visitées localement (sur l'ordinateur de l'utilisateur), on parle alors de cache du navigateur. L'inconvénient est que les données risquent de ne plus être d'actualité. Car, entre temps de nouvelles mises à jour de la page se sont produites sur le serveur. Il est possible de forcer l'ordinateur serveur à commander au navigateur de ne pas prendre les données dans une mémoire cache mais du serveur d'origine. Ceci est recommandé dans les pages actualisées fréquemment.

La forme générale est la suivante : `<meta http-equiv="expires" content="valeur">`. La valeur de l'attribut **content** varie selon le tableau ci-dessous :

content	Description
"0"	Recharger la page du serveur à chaque fois qu'elle est demandée par l'utilisateur.
"n" : Entier > 0	La page doit être rechargée à partir du serveur après n secondes.
"Day, JJ Mon AAAA HH:MM:SS GMT"	A l'arrivée de la date et l'heure (mentionnées dans un format international) la page doit être rechargée. Le jour de semaine et le mois doivent être donnés avec les trois premières lettres de l'appellation en anglais. Par exemple : <code>content="Thu, 01 Jan 2009 05:00:00 GMT"</code> .

A l'aide des META, il est possible aussi de lancer une autre page (rediriger) après un certain délai. La forme générale est : `<meta http-equiv="refresh" content="n; URL=http://adresse_complète">`. Où n est un nombre entier positif ou nul et après **URL=** on peut donner l'adresse en respectant la syntaxe des fichiers. Si le « ; » ainsi que ce qui le suit ne sont pas rajoutés, c'est la page actuelle qui est rechargée.

Remarque :

Il ne faut pas trop abuser avec les balises META. Ceci risque de rendre la page trop volumineuse et lourde à télécharger au niveau de l'utilisateur. Il faut savoir que ce n'est pas uniquement les META qui permettent de bien classer la page dans les résultats des moteurs de recherche. Le contenu de la page joue un rôle très important.

3.2 Le corps

Le corps de la page HTML est défini par la balise `<body>`. La balise `</body>` marque sa fin. La plupart des éléments insérés dans cette partie apparaissent sur le navigateur. Ainsi, cette partie contient tous les éléments à afficher : textes, images, ...

Il faut savoir que dans cette partie, le formatage manuel (c'est à dire l'insertion des sauts de lignes, des espaces supplémentaires...dans le code) est toujours ignoré. Nous verrons plus loin comment insérer ce genre d'éléments.

3.3 Le Commentaire

Comme dans tous les langages informatiques, on peut insérer des commentaires HTML. Pour cela, il y a aussi une balise. Le commentaire peut être inséré n'importe où dans la page mais n'est pas pris en considération (affiché) par le navigateur. Le texte du commentaire doit être inséré entre `<!--` et `-->`.

Par exemple : `<!-- ... le commentaire ... -->`

Il faut savoir que le commentaire peut être sur une ou plusieurs lignes et peut contenir des balises HTML, mais on ne peut pas imbriquer les commentaires.

4. Propriétés de la page

Plusieurs propriétés de la page peuvent être fixées à l'aide des attributs de la balise `<body>`. Ces attributs sont facultatifs.

4.1 Le fond

On peut choisir une couleur de fond pour la page à l'aide de l'attribut **bgcolor**. La valeur de cet attribut obéit à la syntaxe des couleurs (expliquée en début du chapitre).

Il est possible d'insérer une image de fond sur la page à l'aide de l'attribut **background**. Sa valeur obéit à la syntaxe des fichiers. Par défaut l'image de fond défile avec le texte.

Conseil : Il faut toujours choisir une couleur de fond qui se rapproche de la couleur dominante de l'image de fond. Ceci permettra à l'utilisateur de visualiser le contenu de la page en cas où l'image de fond ne s'affiche pas.

4.2 Couleurs par défaut des textes

On peut fixer la couleur par défaut du texte (qu'il soit simple ou lien) comme suit :

- La couleur par défaut du texte : On rajoute l'attribut **text**.
- La couleur par défaut des liens : On rajoute l'attribut **link**.
- La couleur par défaut des liens visités : On rajoute l'attribut **vlink**.
- La couleur par défaut des liens actifs : On rajoute l'attribut **alink**.

Les valeurs de tous ces attributs obéissent à la syntaxe de la couleur.

4.3 Direction du texte

Par défaut, l'écriture du texte est dirigée de gauche à droite. Pour modifier la direction on rajoute l'attribut **dir="rtl"** pour une direction de droite à gauche (page contenant du texte en Arabe, par exemple).

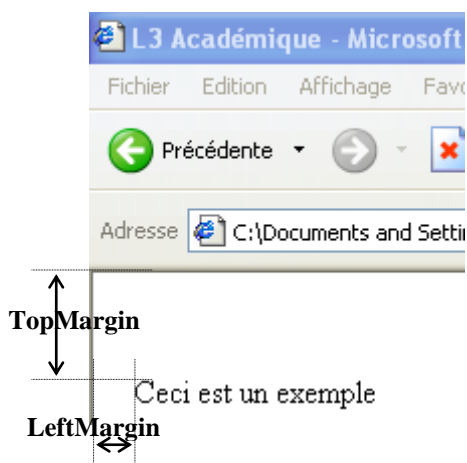
4.4 Les marges de la page

Il s'agit de l'espace qui sera laissé entre le début de la zone d'affichage du navigateur et le premier élément à afficher sur la page. Cet espace est calculé en nombre de pixels.

- La marge du haut : à l'aide de l'attribut **topmargin**.
- La marge gauche : à l'aide de l'attribut **leftmargin**.

La valeur des deux attributs est un entier positif ou nul.

Exemple : `<body topmargin="50" leftmargin="20">` Le résultat est illustré par la figure ci-contre.



5. Mise en forme du texte

5.1 Principales mises en forme

Pour définir la police, la taille et/ou la couleur d'un texte on le met entre **** et ****. Comme suit :

` ... le texte ... `

On peut définir un, deux ou tous les attributs pour cette balise. Le tableau ci-dessous donne plus de précisions :

Attribut	Valeur
face	Il permet d'attribuer le nom de la police de caractères qui sera appliquée au texte. Il est possible d'attribuer plusieurs noms de polices de caractères séparés par des virgules. Ceci est utilisé pour définir une police de remplacement (si la première n'est pas disponible sur la machine de l'utilisateur la suivante la remplace). Par exemple : face="Times New Roman" ou face="Verdana, Georgia" .
color	Il s'agit de la couleur qui sera attribuée au texte. Sa valeur obéit à la syntaxe de la couleur.
size	Il permet de définir la taille du caractère et peut prendre les valeurs de 1 à 7.

Le tableau ci-contre, donne les correspondances entre la taille du caractère en HTML (attribut **size**) et la taille en points utilisée dans les logiciels de traitement de texte.

En plus de la balise ****, il existe d'autres balises de mise en forme du texte pour obtenir comme par exemple du texte en gras, en italique, ... La description de ces balises est donnée dans le tableau suivant :

Size	Equivalent traitement de texte
"1"	8 points
"2"	10 points
"3"	12 points
"4"	14 points
"5"	18 points
"6"	24 points
"7"	36 points

Balises	Définition
<code> ...txt... </code> ou <code> ...txt... </code>	Les deux permettent de mettre le texte en gras .
<code> ...txt... </code> ou <code><i> ...txt... </i></code>	Les deux permettent de mettre le texte en <i>italique</i> .
<code><u> ...txt... </u></code>	Elle permet de souligner un texte.
<code><sup> ...txt... </sup></code>	Elle met le texte en ^{exposant} .
<code><sub> ...txt... </sub></code>	Elle met le texte en _{indice} .

5.2 Mises en forme particulières

Il existe aussi d'autres mises en forme (particulières) données par le tableau suivant:

Balise	Description
<code><h1> ... </h1></code>	Permet d'insérer un titre de niveau <i>i</i> . La valeur de <i>i</i> va de 1 à 6.
<code><code> ... </code></code>	Pour insérer un code.
<code><kbd> ... </kbd></code>	Pour insérer un texte en format clavier.
<code><var> ... </var></code>	Pour insérer une variable (mathématique).
<code><cite> ... </cite></code>	Pour insérer une citation.
<code><dfn> ... </dfn></code>	Pour insérer une définition.
<code><samp> ... </samp></code>	Pour insérer un exemple.
<code><blockquote> ... </blockquote></code>	Affiche le texte mis entre ses balises sous forme de bloc en retrait.
<code><address> ... </address></code>	Pour insérer une adresse.
<code><tt> ... </tt></code>	Imprimé (style machine à écrire).
<code><pre> ... </pre></code>	Texte pré formaté. Les espaces, les sauts et même les balises apparaîtront tels qu'ils ont été écrits entre <code><pre></code> et <code></pre></code> .
<code><big> ... </big></code>	Pour agrandir de +1 la taille du texte.
<code><small> ... </small></code>	Pour réduire de -1 la taille du texte.

5.3 Remarques

- Il faut faire attention à la fermeture de balises : ordre « LIFO / Pile ».
- En cas de conflit entre balises, la balise la plus proche sera appliquée.
- Il y a d'autres balises comme `<blink>` (texte clignotant) et `<strike>` (texte barré) qui sont déconseillées.

5.4 Exemple

Soit la partie `<body>` d'un code HTML ci-dessous (coté gauche) qui donne l'affichage ci-dessous (à droite) :

```

<h1>Titre de niveau 1</h1>
<h2>Titre de niveau 2</h2>
<h3>Titre de niveau 3</h3>
<h4>Titre de niveau 4</h4>
<h5>Titre de niveau 5</h5>
<h6>Titre de niveau 6</h6>
<font size="3">
<code>code</code><br>
<kbd>clavier</kbd><br>
<var>variable</var><br>
<cite>citation</cite><br>
<dfn>définition</dfn><br>
<samp>exemple</samp><br>
On va insérer <blockquote> un petit bloc en retrait
</blockquote>pour jouer le rôle de citation
<address>adresse</address>
<tt> imprimé ...</tt>
<pre> texte
préformaté ... </pre>
texte normalement de taille 3 :
<big> texte agrandi </big>,
<small>texte réduit</small>
</font>

```

Titre de niveau 1

Titre de niveau 2

Titre de niveau 3

Titre de niveau 4

Titre de niveau 5

Titre de niveau 6

code
clavier
variable
citation
définition
exemple
On va insérer

un petit bloc en retrait

pour jouer le rôle de citation

adresse

imprimé ...

texte
préformaté ...

texte normalement de taille 3 : texte agrandi , texte réduit

5.5 Insertion des caractères spéciaux

Certains caractères ne s'affichent pas comme les symboles réservés du HTML (le < et le >), les espaces supplémentaires ou les caractères accentués non reconnus par tous les navigateurs. Les tableaux ci-dessous donnent les principaux codes HTML associés aux symboles et caractères spéciaux.

Tableau de quelques caractères accentués :

Caractère	Code textuel	Code numérique
À	â	â
Â	Â	Â
Á	à	à
À	À	À
Ç	ç	ç
Ç	Ç	Ç
É	é	é
É	É	É
Ê	ê	ê
Ê	Ê	Ê
È	è	è
È	È	È
Ë	ë	ë
Î	î	î
Ï	ï	ï
Ô	ô	ô
Ö	ö	ö
œ	œ	œ
Û	û	û
Ü	ü	ü
U	ù	ù

Tableau de quelques symboles :

Caractère	Code textuel	Code numérique
Espace	 	
<	<	<
>	>	>

≤	≤	≤
≥	≥	≥
≠	≠	≠
&	&	&
"	"	"
'	´	´
©	©	©
®	®	®
™	™	™

Tableau des symboles de monnaies :

Caractère	Code textuel	Code numérique
€	€	€
£	£	£
¥	¥	¥

Tableau de quelques caractères de l'alphabet grecque :

Caractère	Code textuel	Code numérique
α	α	α
β	β	β
γ	γ	γ
Δ	Δ	Δ
λ	λ	λ
μ	μ	μ
π	π	π
Σ	Σ	Σ
φ	φ	φ
Ω	Ω	Ω
ψ	ψ	ψ

6. Les séparateurs

6.1 Les paragraphes

Un paragraphe est défini par <p>. La balise fermante </p> est facultative. Pour aligner le paragraphe, on rajoute l'attribut **align** qui peut avoir les valeurs : "center", "left" ou "right". L'insertion d'un paragraphe définit une nouvelle ligne (saut) avec interligne.

6.2 Saut de ligne

Il s'agit d'un retour à la ligne suivante (tout en restant dans le même paragraphe) sans interligne. Pour réaliser ce saut, on utilise la balise
. Cette balise n'a pas de balise fermante.

6.3 Barre horizontale

L'insertion d'une ligne (barre) horizontale se fait à l'aide de la balise <hr>. Cette balise n'a pas de balise fermante. Elle définit automatiquement un nouveau paragraphe indépendant de ce qui précède et de ce qui suit. Cette balise admet plusieurs attributs résumés par le tableau ci-dessous :

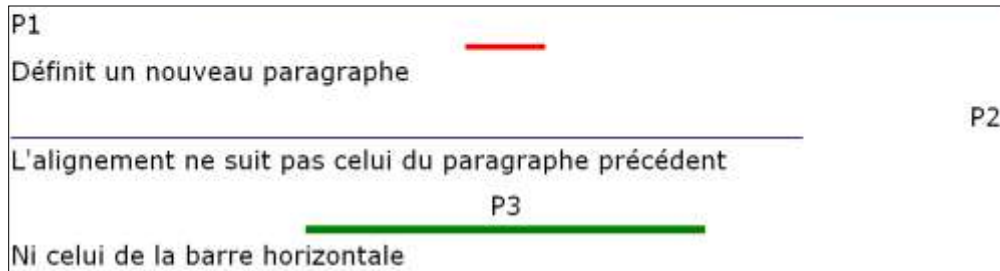
Attribut	Description
align	Permet de définir l'alignement de la barre, obéit à la syntaxe de l'alignement horizontal.
width	Fixe la largeur de la barre. Sa valeur obéit à la syntaxe des tailles (voir début du chapitre)
size	Sa valeur est un nombre entier positif. Permet de fixer l'épaisseur de la barre en pixels.
color	Sa valeur obéit à la syntaxe des couleurs. Elle permet de fixer la couleur de cette ligne.

Exemple :

Soit la portion de code HTML suivante :

```
<p align="left">P1
<hr width="100" size="7" noshade color="#FF0000">
Définit un nouveau paragraphe
<p align="right">P2
<hr align="left" width="80%" color="#000080">
L'alignement ne suit pas celui du paragraphe précédent
<p align="center">P3
<hr width="500" size="10" color="green">
Ni celui de la barre horizontale
```

Ceci donne l'affichage suivant :



7. Les listes

7.1 Listes numérotées automatiquement

Les listes ordonnées (numérotée) sont insérées à l'aide de `` (pour marquer le début de la liste) et `` (pour marquer la fin de la liste). Chaque élément de la liste doit être précédé par la balise ``. La balise fermante `` est facultative.

Exemple : Soit la portion de code ci-contre (à gauche) qui donne l'affichage ci-contre (à droite) :

```
<ol>
<li>HTML
<li>PHP
<li>MYSQL
<li>CSS</li>
<li>Javascript
</ol>
```

```
1 HTML
2 PHP
3 MYSQL
4 CSS
5 Javascript
```

7.1.1 Type de numérotation

La numérotation par défaut est : 1, 2, ... etc (chiffres arabes à partir de 1). Il est possible de changer la façon de numéroté à l'aide de l'attribut **type** dans ``. Le tableau ci-dessous donne les valeurs possibles pour cet attribut :

Type	Description
"I"	Numérotation en grands chiffres romains : I II III ...
"i"	Numérotation en petits chiffres romains : i ii iii ...
"A"	Numérotation alphabétique majuscules : A B C ...
"a"	Numérotation alphabétique minuscules : a b c ...
"1"	Numérotation ordinaire : 1 2 3 ...

Exemple : Soit la portion de code suivante qui donne l'affichage ci-dessous à droite.

```
<ol type="I">
<li>HTML</li>
<li>PHP</li>
<li>MYSQL
<li>CSS</li>
<li>Javascript
</ol>
```

```
I HTML
II PHP
III MYSQL
IV CSS
V Javascript
```

7.1.2 Début de numérotation

Par défaut la numérotation commence à partir de 1, a, A, i ou I. (Selon l'attribut **type**) Pour changer le numéro du premier élément on utilise l'attribut **start** dans la balise ****. La valeur de **start** est un entier "*n*" et ça commence de :

- *n* si la numérotation, est ordinaire.
- La valeur correspondante à *n* en chiffres romains, si le **type** est "I" ou "i".
- La *n*^{ème} lettre de l'alphabet, si le **type** est A ou a.

7.2 Les listes à puce

Les listes non ordonnées (à puces) sont insérées de la même manière que les listes ordonnées sauf qu'on utilise **** (pour marquer le début de la liste) et **** (pour marquer la fin de la liste) au lieu de **** et ****.

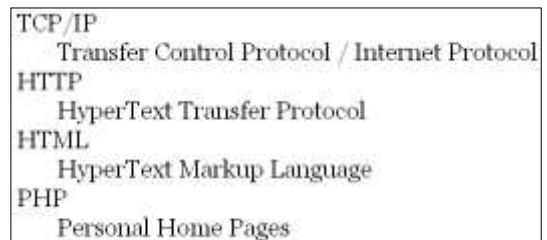
Il est possible de changer la forme de la puce à l'attribut **type** dans **** (pour toute la liste) ou dans **** (pour un élément). Les valeurs possibles sont : **square**, **disc**, **circle**. On peut aussi importer une image pour jouer le rôle de la puce avec l'attribut **imagesrc** dans ****. Sa valeur obéit à la syntaxe des fichiers. Actuellement, la majorité des navigateurs ne reconnaissent plus ces attributs, qui sont gérés par les CSS.

7.3 Les listes de définitions

Les listes de définitions sont utilisées pour afficher des glossaires ou dictionnaires. On utilise **<dl>** et **</dl>** pour marquer le début et la fin de la liste. Chaque élément (un titre) doit être encadré par **<dt>** et **</dt>**. La définition de l'élément doit être encadrée par **<dd>** et **</dd>**. Il faut savoir que **</dt>** et **</dd>** sont facultatives.

Exemple : Soit la portion de code suivante qui donne l'affichage ci-dessous (à gauche) qui donne l'affichage ci-dessous (à droite) :

```
<dl>
  <dt>TCP/IP
  <dd>Transfer Control Protocol/Internet Protocol
  <dt>http
  <dd>HyperText Transfer Protocol
  <dt>HTML
  <dd>HyperText Markup Language
  <dt>PHP
  <dd>Personal Home Pages
</dl>
```



Remarque :

On peut imbriquer les listes, un élément **** (ou même **<dd>**) peut contenir une liste ****, **** ou **<dl>**.

Exemple : Soit la portion de code suivante qui donne l'affichage ci-contre :

```
<ol type="I">
  <li>HTML
    <ul>
      <li>Balise
      <li>Mise en forme du texte
      <li>Les tableaux
      <li>Les formulaires
    </ul>
  </li>
  <li>PHP
    <ul>
      <li>Insertion dans un code HTML
      <li>Variables PHP
      <li>Instructions PHP
    </ul>
  </li>
</ol>
```



8. Les liens

Il y a deux types de liens (appelés aussi références) qu'on peut définir sur un texte ou une image : internes et externes.

8.1 Les liens internes :

Ils désignent un point précis, repéré par une balise, dans le corps même du document. En d'autres termes, ça va générer un déplacement dans la même page. La réalisation d'un lien interne se fait en deux étapes : la définition de l'étiquette et la définition de la référence. On fixe un endroit dans le document (la cible).

On définit une étiquette à cet endroit (une ancre) comme suit : `...`. On peut ne rien mettre entre `<a>` et ``.

On définit le lien (la référence) sur un texte ou une image comme suit : `...`. En cliquant sur ce texte ou cette image, il y aura déplacement dans la page (sur le navigateur) vers l'endroit étiqueté.

8.2 Les liens externes :

Ils spécifient simplement une autre page HTML, un document multimédia ou l'accès à un service comme le e-mail ou le FTP. On définit le lien sur un texte ou une image comme suit : `...`. La valeur de `href` obéit à la syntaxe de fichier, comme elle peut spécifier un autre service si la valeur de `href` est :

- `"ftp://url"` : va vers un le serveur ftp spécifié par l'url.
- `"mailto:adr_email?subject=objet_message"` : l'application mail est lancée pour l'envoi d'un email à l'adresse définie par `adr_email` avec le champ objet contenant la chaîne de caractères `objet_message`. On peut omettre l'objet en écrivant `"mailto:adr_email"`.

Remarques :

- On peut combiner lien externe avec interne avec `href="fichier#etiquette"`.
- D'autres propriétés des liens seront données dans la partie de ce chapitre qui traite des cadres.

9. Insertion d'une image

L'insertion d'une image se fait à l'aide de la balise ``. L'image ne fait pas partie du code HTML. Elle est importée à partir d'un fichier externe. C'est pour cette raison qu'il faut spécifier le nom du fichier image. Ainsi, l'attribut `SRC` est obligatoire et sa valeur obéit à la syntaxe de fichier. Par exemple : ``. L'image est affichée à sa dimension réelle.

9.1 Taille de l'image

On peut spécifier la dimension (hauteur, largeur) de l'image à l'aide des attributs `width` et `height`. Leurs valeurs sont des entiers positifs qui correspondront à la hauteur et à la largeur de l'affichage de l'image en pixels. En fixant ces deux attributs, il faut faire très attention à la déformation de l'image.

9.2 Texte alternatif

On peut définir un texte alternatif à l'aide de l'attribut `alt`. Sa valeur est texte libre. Le rôle du texte alternatif est d'apparaître à la place de l'image en cas où l'image ne peut pas être affichée : fichier inexistant, connexion trop lente, ...

9.3 Bordure et espacement

On peut définir la bordure autour de l'image à l'aide de l'attribut `border`. Sa valeur est un entier (positif ou nul) qui correspondra à l'épaisseur en pixels.

On peut aussi définir un espace (vide) autour de l'image. La balise `` dispose de deux attributs d'espacement. L'attribut `hspace` permet de définir un espacement horizontal : par rapport aux éléments à droite et à gauche de l'image. L'attribut `vspace` permet de définir un espacement vertical : par rapport aux éléments en haut et en bas de l'image. Leurs valeurs sont des entiers positifs ou nuls en pixels.

9.4 Alignement

L'attribut d'alignement de la balise `` concerne particulièrement le texte qui est près de l'image. Il s'agit de l'attribut **align** dont les valeurs sont : "left", "right", "center", "top", "bottom" ou "middle". Le tableau ci-dessous explique chacune des valeurs :

ALIGN	Explication
"left"	L'image est mise complètement à gauche par rapport au texte qui est près d'elle.
"right"	L'image est mise complètement à droite par rapport au texte qui est près d'elle.
"center"	L'image est centrée et le texte qui est près d'elle va l'entourer.
"top"	Le texte de la même ligne que l'image s'affiche en haut et laisse un vide en bas de l'image : décalage par rapport à la ligne suivante.
"bottom"	Le texte de la même ligne que l'image s'affiche en bas et laisse un vide en bas de l'image : décalage par rapport à la ligne précédente.
"middle"	Le texte de la même ligne que l'image s'affiche en haut par rapport à l'image et laisse un vide en haut et en bas de l'image : décalage par rapport aux lignes précédente et suivante.

9.5 Remarque

- Les extensions autorisées des fichiers images sont : *.gif .jpg .jpeg* et *.png*.
- En insérant une image, il faut faire attention aux droits d'auteurs. Il faut utiliser ses propres images ou les images libres de droit. Il faut aussi prendre en considération la liberté individuelle (protection de la vie privée des personnes) et la propriété des biens.
- Il est conseillé de donner à **width** et **height** des valeurs égales à la largeur et la hauteur réelles de l'image.
- Si on veut utiliser une miniature ou un agrandissement de l'image, il vaut mieux la miniaturiser ou l'agrandir dans un logiciel de traitement d'images et insérer la nouvelle image dans la page.
- De préférence, il faut éviter d'utiliser des images de grande taille (dimension et poids) dans la page Web.
- On peut utiliser une petite image compressée sur la page et faire un lien vers l'image de grande qualité.

10. Les tableaux

Les tableaux sont très importants dans la conception et la mise en forme d'une page Web. Ils permettent de bien placer chaque élément dans la page. Ainsi, une excellente mise en page, une parfaite maîtrise de l'espace et une bonne structuration du contenu sont possibles.

Un tableau est un ensemble de lignes et de colonnes. Un élément d'une ligne donnée *i* et d'une colonne donnée *j* est appelé cellule. On utilise les balises `<table>` et `</table>` pour marquer le début et la fin du tableau.

Un tableau HTML est organisé en lignes (une suite de plusieurs lignes) et chaque ligne contient un ensemble de cellules. Il faut s'assurer d'insérer le même nombre de cellules dans chaque ligne.

10.1 Lignes et Cellules d'un tableau :

Le début d'une ligne du tableau (Table Row) est marqué par la balise `<tr>`. La fin de la ligne est marquée par `</tr>` mais cette balise n'est pas obligatoire.

A l'intérieur d'une ligne on peut définir une ou plusieurs cellules. Le début de chaque cellule est marqué par la balise `<td>` (Table Data). La balise de fin de cellule `</td>` n'est pas obligatoire.

Il est possible de définir des cellules spéciales à l'aide des balises `<th>` et `</th>`. Ces cellules seront considérées comme des titres de colonne. On peut s'en passer de ce genre de cellules.

10.2 Titre du tableau

On peut définir un élément spécifique qui jouera le rôle de titre du tableau à l'aide des balises `<caption>` et `</caption>` qu'il faut insérer juste après `<table>`. Le titre du tableau n'est pas obligatoire.

10.3 Propriétés du tableau

10.3.1 Alignement du tableau

On encadre le tableau par les balises `<DIV>` (juste avant `<table>`) et `</div>` (juste après `</table>`) avec comme attribut `align`. On peut aussi le faire à l'aide de l'attribut `align` de la balise `<table>`. Les valeurs possibles de `align` sont les valeurs de l'alignement horizontal vues au début de ce chapitre.

10.3.2 Largeur et Hauteur

Pour fixer la largeur et la hauteur du tableau, on rajoute les attributs `width` et `height` à la balise `<table>`. Leurs valeurs obéissent à la syntaxe des valeurs de la taille (voir début du chapitre) : en absolu (pixels) ou en relatif (%).

10.3.3 Bordure

Le tableau ci-contre explique les attributs de `<table>` permettant de paramétrer la bordure du tableau.

La valeur du premier attribut est un entier positif ou nul. Les valeurs des autres attributs obéissent à la syntaxe des valeurs de couleurs. Les deux derniers attributs sont utilisés pour donner un effet 3D (ombrage) à la bordure.

Attribut	Description
<code>border</code>	Pour fixer l'épaisseur de la bordure en pixels.
<code>borderColor</code>	Pour définir la couleur principale de la bordure.
<code>borderColorLight</code>	Pour définir la couleur de la bordure claire.
<code>borderColorDark</code>	Pour définir la couleur de la bordure sombre.

10.3.4 Fond

On définit le fond de tout le tableau par une couleur et/ou une image. La balise `<table>` a deux attributs qui permettent de définir le fond comme l'explique le tableau suivant :

Attribut	Description
<code>b bgcolor</code>	Pour définir la couleur de fond. Sa valeur obéit à la syntaxe des valeurs de couleurs.
<code>Background</code>	Pour définir une image de fond. Sa valeur obéit à la syntaxe des fichiers.

10.3.5 Espacements

Les espacements à l'intérieur du tableau sont fixés à l'aide des attributs `cellspacing` et `cellpadding` de `<table>` dont les valeurs sont des entiers positifs ou nuls. Le premier pour définir l'espacement en pixels entre deux cellules adjacentes (voisines) et le second pour fixer la marge interne en pixels des cellules.

Exemple :

Soit la portion de code HTML ci-contre :

```
<div align="center">
  <!-- tableau aligné au centre -->
  <table width="450" height="425" border="5"
    cellspacing="10" cellpadding="25" bordercolor="black">
    <!-- Ligne 1 -->
    <tr>
      <!-- 3 cellules titres TH dans la première ligne -->
      <th>Année <!-- 1ère cellule de la première ligne -->
      <th>Effectif <!-- 2ème cellule de la première ligne -->
      <th>Évolution <!-- 3ème cellule de la première ligne -->
    <!-- Ligne 2 -->
    <tr>
      <!-- 3 cellules ordinaires TD -->
      <td>2000 <td>18 000 <td>12%
    <!-- Ligne 3 -->
    <tr>
      <td>2002 <td>20 000 <td>10 %
    <!-- Toujours 3 cellules dans chaque ligne -->
    <tr>
      <td>2004 <td>21 000 <td>5 %
    <!-- Ligne 5 -->
    <tr>
      <td>2006 <td>21 500 <td>2.5 %
    </table>
  <!-- Fin du tableau -->
</div>
```

Ce code donne l'affichage suivant :

The diagram shows a table with 5 rows and 3 columns. The first row contains headers: 'Année', 'Effectif', and 'Évolution'. The subsequent rows contain data: (2000, 18 000, 12%), (2002, 20 000, 10%), (2004, 21 000, 5%), and (2006, 21 500, 2.5%). Annotations indicate the table's overall width is 450, height is 425, cellpadding is 25, cellspacing is 10, and border is 5.

Année	Effectif	Évolution
2000	18 000	12%
2002	20 000	10%
2004	21 000	5%
2006	21 500	2.5%

10.4 Propriétés des cellules

10.4.1 Alignement dans la cellule

On peut fixer l'alignement des éléments insérés à l'intérieur d'une cellule d'un tableau à l'aide des attributs **align** et **valign** de **<td>** (ou **<th>**). Le premier définit l'alignement horizontal ("**left**", "**right**" ou "**center**") et le second l'alignement vertical ("**top**", "**bottom**" ou "**middle**"). Il est possible d'utiliser ces attributs dans la balise de ligne **<tr>** pour pouvoir les appliquer sur toutes les cellules de la ligne et éviter les redondances.

10.4.2 Largeur et Hauteur

On utilise dans **<td>** (ou **<th>**) les mêmes attributs que pour **<table>** : **width** et **height**. Leur utilisation est la même, mais cette fois-ci les valeurs vont s'appliquer uniquement à la cellule concernée. Si la valeur est en pourcentage, la cellule aura une largeur (ou hauteur) égale à ce pourcentage par rapport à la largeur du tableau. Il est possible d'utiliser ces attributs dans la balise de ligne **<tr>**. Il est même conseillé d'utiliser l'attribut **height** plutôt dans **<tr>**.

10.4.3 Le fond

De la même façon que pour **<table>**, on peut utiliser les attributs **b bgcolor** et **background** dans **<td>**, **<th>** ou **<tr>**. Seulement le fond des cellules concernées qui change.

10.4.4 Couleur bordure

De la même façon que pour **<table>**, on peut utiliser les attributs **bordercolor**, **bordercolorlight** et **bordercolordark** dans **<td>**, **<th>** ou **<tr>**. Seulement la bordure de la cellule concernée qui change.

10.4.5 Éviter le renvoi automatique à la ligne

Pour afficher les données de la cellule sur la même ligne (interdire le renvoi à la ligne), on a l'attribut sans valeur **nowrap** qu'il faut rajouter à la balise **<td>** ou **<th>**.

10.5 Fusionnement des cellules

On peut fusionner des cellules adjacentes horizontalement ou verticalement à l'aide des attributs **colspan** et **rowspan** dans la balise **<td>**.

10.5.1 Fusionnement horizontal

On utilise l'attribut **colspan** dont la valeur est un entier $n \geq 2$ qui correspond au nombre de cellules fusionnées. Les $n-1$ autres cellules sur la même ligne doivent être supprimées.

10.5.2 Fusionnement vertical

On utilise l'attribut `rowspan` dont la valeur est un entier $n \geq 2$ qui correspond au nombre de cellules fusionnées. Une cellule sur les $n-1$ lignes suivantes (sur la même colonne) doit être supprimée.

10.5.3 Exemple

Soit la portion de code HTML ci-contre :

```
<!-- On va définir un tableau de 3 lignes 4 colonnes de largeurs identiques -->
<table border="1" width="200" bordercolor="black">
<!-- 1ère ligne -->
  <tr>
    <td colspan="3">1 1
<!-- cette cellule représente 3 cellules adjacentes sur la même ligne
    On supprime les deux cellules suivantes sur cette ligne      -->
<!-- Faire attention à la largeur et hauteur lors de la fusion Ici, on n'a pas mis l'attribut width
    dans cette cellule. Si on le rajoute, il faut le mettre à 75% (la somme des 3 largeurs)      -->
<!-- A supprimer : fusionnée dans 1 1    <td width="25%">1 2 -->
<!-- A supprimer : fusionnée dans 1 1    <td width="25%">1 3 -->
    <td width="25%">1 4
<!-- 2ème ligne -->
  <tr>
    <td rowspan="2" width="25%">2 1
<!-- cette cellule représente 2 cellules adjacentes sur la même colonne
    On supprime la cellule de la 1ère colonne de la ligne suivante      -->
    <td width="25%">2 2
    <td colspan="2" rowspan="2">2 3
<!-- cette cellule représente 2 cellules adjacentes sur la même ligne et 2 cellules adjacentes sur
    la même colonne. Elle représente la fusion de 4 cellules.
    On supprime 3 cellules : une sur cette ligne et deux sur la ligne suivante      -->
<!-- A supprimer : fusionnée dans 2 3    <td width="25%">2 4-->
<!-- 3ème ligne -->
  <tr>
<!-- A supprimer : fusionnée dans 2 1    <td width="25%">3 1 -->
    <td width="25%">3 2
<!-- A supprimer : fusionnée dans 2 3    <td width="25%">3 3 -->
<!-- A supprimer : fusionnée dans 2 3    <td width="25%">3 4 -->
</table>
```

Ce code donne l'affichage ci-contre :

11		14
21	22	23
	32	

11. Frames – Cadres :

Les frames (cadres) permettent de définir des zones d'affichage dans la page Web. Chaque frame correspond à une page HTML en plus de la page des cadres.

11.1 Page des Cadres

La syntaxe de la page des cadres est particulière. Le corps de la page est défini par la balise `<FRAMESET>` et `</FRAMESET>` au lieu de `<body>`.

La balise `<FRAMESET>` définit une décomposition (horizontale ou verticale) en 2 ou plusieurs cadres. Pour définir une décomposition horizontale, on utilise dans `<FrameSet>` l'attribut `COLS`. Pour définir une décomposition verticale, on utilise dans `<FrameSet>` l'attribut `ROWS`. Seulement un des deux attributs doit être utilisé.

La valeur de ces deux attributs est une suite d'éléments séparés par des virgules. Chaque élément représente la hauteur d'une ligne ou la largeur d'une colonne : espace réservé au cadre. Un élément (ligne ou colonne) peut être un cadre défini par la balise `<FRAME>` ou un ensemble de cadres défini par la balise `<FRAMESET>`. Ainsi, entre `<Frameset>` et `</Frameset>`, on doit définir tous les cadres dont le nombre est égal au nombre d'éléments dans la

valeur de l'attribut **COLS** ou **ROWS**. Chaque cadre est défini par la balise **<Frame>** comme on peut avoir au lieu d'un cadre, une autre décomposition à l'aide d'un autre **<Frameset>**.

L'espace réservé (hauteur ou largeur) est défini par :

- Le nombre de pixels (entier > 0)
- Un pourcentage (entier > 0 suivi de %)
- * : pour utiliser tout l'espace restant.

Par exemple : **<Frameset rows="100,*,150,10%">**. Ceci définit une décomposition de la page en 4 lignes. La première ligne aura une hauteur de 100 pixels, la troisième de 150 pixels, la quatrième de 10% de la hauteur de la zone d'affichage de la fenêtre du navigateur et la deuxième occupera l'espace restant sur la zone d'affichage du navigateur.

11.2 Propriétés des Cadres

Le tableau ci-dessous résume brièvement les propriétés de la page des cadres :

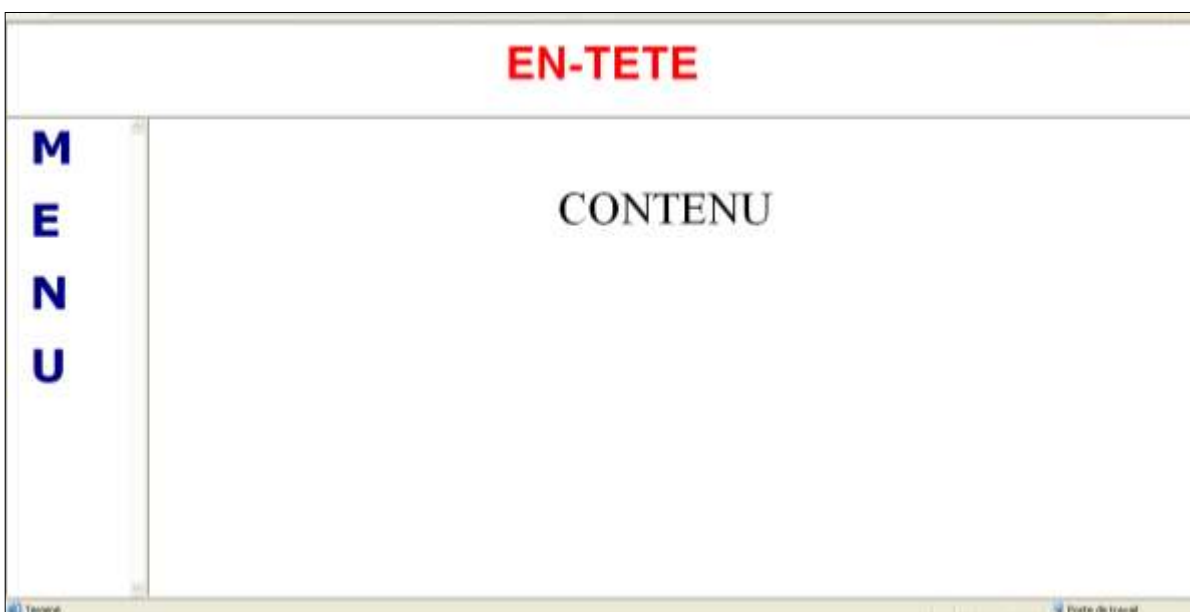
Balise	Attribut	Valeur	Description
<Frameset>	FrameSpacing	entier≥0	Espacement entre cadres
<Frame>	Name	libre	Nom du cadre
<Frame>	SRC	fichier	Page HTML associée au cadre
<Frame>	Target	nom d'un cadre.	Cadre d'ouverture par défaut des liens. Le nom du cadre doit exister.
<Frame>	Scrolling	auto, yes ou no	Définit si les barres de défilements doivent être affichées (yes) ou non (no) ou selon le contenu du cadre (auto)
<Frame>	Noresize	Sans valeur	Interdire de redimensionner le cadre.
<Frame>	MarginHeight	entier≥0	Marges internes du cadre en hauteur (haut et bas).
<Frame>	MarginWidth	Entier≥0	Marges internes (à droite et à gauche) du cadre.

11.3 Page de remplacement

Il existe des navigateurs qui ne reconnaissent pas les cadres. La solution est d'insérer juste avant la dernière **</frameset>** le corps de la page qui s'affiche au lieu des cadres entre **<noframes>** et **</noframes>**.

11.4 Exemple

On veut obtenir des cadres comme le montre le schéma ci-dessous. On a 2 lignes de cadres et la 2ème ligne composée de 2 colonnes de cadres.



Le code HTML correspondant est le suivant :

```
<html>
  <head>
    <title>Les cadres</title>
  </head>
  <frameset framespacing="0" rows="100,*">
    <!-- la première ligne de hauteur 100 pixels est un cadre -->
    <frame name="haut" scrolling="no" noresize target="contenu" src="haut.htm">
    <!-- Un seul noresize bloquera tous les autres pour ce cas -->
    <!-- la seconde ligne qui occupera le reste de la fenêtre est
    une décomposition en 2 colonnes cadres -->
    <frameset cols="150,*" framespacing="0">
      <!-- la première colonne de largeur 150 pixels est un cadre -->
      <frame name="menu" marginwidth="25" target="contenu" scrolling="yes" src="menu.htm">
      <!-- la seconde colonne occupera le reste de la largeur est un cadre -->
      <frame name="contenu" marginheight="20" scrolling="auto" src="contenu.htm">
    </frameset>
    <!-- Si le navigateur ne reconnaît pas les cadres -->
    <noframes>
      <body topmargin="5" leftmargin="5">
        <p>Cette page utilise des cadres, mais votre navigateur ne les prend pas en
charge.</p>
      </body>
    </noframes>
  </frameset>
</html>
```

11.5 Liens dans les Cadres

Il faut définir le cadre cible où s'ouvrira le lien. Par défaut, le lien s'ouvrira dans le cadre spécifié par **Target** dans la balise **<FRAME>**, mais on peut le changer. Ceci est réalisable en rajoutant à la balise **<a>** l'attribut **TARGET** avec le nom du cadre destination comme valeur. Comme on peut utiliser certaines valeurs prédéfinies de l'attribut **target** résumées par le tableau ci-dessous :

Valeur	Description
_self	Pour ouvrir le lien dans le même cadre.
_parent	Pour ouvrir le lien dans le cadre père.
_top	Pour ouvrir le lien dans la page entière.
_blank	Pour ouvrir le lien dans une nouvelle fenêtre.

Il faut savoir que **_parent** et **_top** ne donnent pas nécessairement la même chose. Par exemple, si le cadre est dans une page de cadres qui est elle aussi cadre.

11.6 A propos des Cadres

Les cadres sont déconseillés pour diverses raisons : le nombre important de pages (fichiers) utilisées pour afficher juste une page ainsi que le problème d'indexation par les moteurs de recherche.

12. Les formulaires :

Un formulaire est un ensemble d'éléments permettant la saisie d'informations. Le HTML s'occupe du côté description seulement. Il ne réalise pas du traitement des informations saisies.

Pour définir un formulaire on utilise les balises : **<form>** et **</form>**. Elles permettent de délimiter les éléments du même formulaire.

12.1 Propriétés du formulaire

On définit les propriétés du formulaire à l'aide des attributs de la balise **<form>**.

12.1.1 Le nom du formulaire

On peut attribuer un nom formulaire à l'aide de l'attribut **name**. Ceci s'avère très utile dans les traitements côté client : JavaScript par exemple.

12.1.2 L'action

Si les données du formulaire doivent être envoyées au serveur pour les traiter, il faut donner le nom de l'application qui récupérera le formulaire. On utilise l'attribut **action** dont la valeur obéit à la syntaxe des fichiers.

12.1.3 Cadre ou fenêtre de la réponse

L'attribut **target** est utilisé de la même façon que pour la balise `<a>` (voir ci-dessus dans la section liens dans les cadres) pour l'affichage de la réponse. C'est à dire dans quel cadre ou dans une nouvelle fenêtre.

12.1.4 La méthode d'envoi des données

L'attribut **method** à 2 valeurs possibles : **get** et **post**. Par défaut (si cet attribut n'est pas spécifié) c'est la méthode **get** qui sera appliquée.

get : Les données sont envoyées avec l'URL. Ceci limite le nombre de caractères envoyés par le formulaire. Les données apparaissent sur la barre d'adresse du navigateur ce qui peut être un inconvénient lors de l'envoi d'un mot de passe. Par contre, cette méthode permet de faire appel à un script juste en tapant l'adresse et les données avec.

post : Les données sont envoyées dans une zone spéciale sur le serveur. Ce qui permet de ne pas limiter la taille des données.

12.2 Éléments des formulaires

Le formulaire est composé d'éléments d'entrée définis par l'une des trois balises : `<input>`, `<textarea>` et `<select>`.

Les trois balises ont un attribut commun : **name**. Il permet de les identifier de manière unique. C'est un attribut obligatoire et important.



Les éléments (champs) `<input>` se distinguent par la valeur de l'attribut **type** :

- **"text"** : Zone (Champ) de texte simple.
- **"password"** : Zone de saisie de mot de passe.
- **"hidden"** : Champ masqué.
- **"radio"** : Case d'option (bouton radio).
- **"checkbox"** : Case à cocher.
- **"file"** : Pour sélectionner un fichier à envoyer.
- **"button"** : Bouton de commande (simple).
- **"submit"** : Bouton d'envoi.
- **"reset"** : Bouton de réinitialisation.

Les champs `<select>` et `<textarea>` sont uniques. C'est à dire chaque balise représente uniquement un élément.

12.2.1 Les champs `<input>` textuels

Il s'agit des champs de **Type** : **"text"**, **"password"** et **"hidden"**.

Type	Description	Illustration
"text"	Champ (zone) de saisie de texte simple permet de saisir un texte comme : nom, prénom, ...	
"password"	Zone de saisie mot de passe permet de faire apparaître à l'écran chaque caractère en *.	
"hidden"	Champ textuel masqué. Permet de transférer des données sans les afficher à l'écran.	



Les champs `<input>` de type **"text"** ou **"password"** peuvent avoir d'autres attributs facultatifs :

- **value** : Texte par défaut qui s'affiche.
- **size** : Largeur du champ en nombre de caractères.
- **maxlength** : Le nombre maximal de caractères qui peuvent être tapés dans ce champ.

Le champ `<input>` de type **hidden** nécessite obligatoirement l'attribut **value**.

12.2.2 Les champs `<input>` cases

Il s'agit des champs de **type** : **"radio"** et **"checkbox"**.

Type	Description	Illustration
"radio"	Case d'option (bouton radio). Permet un choix en exclusion mutuelle (ayant le même nom).	
"Checkbox"	Case à cocher. Pour sélectionner plusieurs options à la fois.	

Ces deux champs ont les mêmes attributs :


- **Value** : La valeur.
- **Checked** : pour sélectionner ou cocher par défaut.

12.2.3 Les champs <INPUT> boutons

Il s'agit des champs de **type** : **button**, **submit** et **reset**.

- **"button"** : Bouton simple, il est utile généralement avec scripts coté client (Javascript, ...)
- **"submit"** : Lance l'envoi des données du formulaire à l'application (sur le serveur) définie par Action.
- **"reset"** : Annule le formulaire en effaçant toutes les données saisies et en remettant les valeurs par défaut.

Les trois champs ont l'attribut **value** qui définit le label (étiquette) du bouton : texte apparaissant sur le bouton. Par contre, l'attribut **name** n'est pas obligatoire pour **"submit"** et **"reset"**. Pour la forme, les trois apparaissent sous

la même forme : 

12.2.4 Les champs <input> fichier

Il s'agit des champs de **type** : **"file"**. Il permet d'envoyer dans le formulaire un fichier à partir de la machine utilisateur vers le serveur. Il affiche une zone textuelle et un bouton de parcours. On peut rajouter les attributs suivants :

- **size** : Comme pour le champs **"text"**, elle définit la taille de la zone textuelle en nombre de caractères.
- **accept** : Permet de définir un filtre. C'est à dire le type de fichier autorisé.



12.2.5 Champ de saisie multi-lignes :

Pour définir une zone de saisie sur plusieurs lignes, on utilise les balises : **<textarea>** et **</textarea>**. Entre les deux balises, on peut mettre le texte qui apparaîtra par défaut. En plus de **Name**, **<textarea>** a d'autres attributs facultatifs :

- **rows** : Sa valeur est un entier positif qui représente le nombre de lignes qu'occupera la zone de saisie.
- **cols** : Sa valeur est un entier positif définissant la largeur de la zone en nombre de caractères.



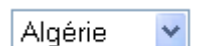
12.2.6 Liste déroulante (zone de sélection)

On utilise les balises **<select>** et **</select>** pour délimiter la liste. En plus de **Name**, **<select>** a deux attributs facultatifs :

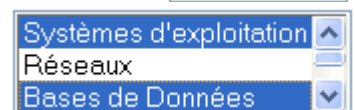
- **size** : Le nombre de lignes qu'occupera la zone.
- **multiple** : Attribut sans valeur, pour autoriser le choix multiple. (En utilisant la touche clavier **CTRL**)

Pour définir les éléments de la liste déroulante, on utilise entre **<select>** et **</select>** la balise **<option>** autant de fois qu'on a d'élément dans la liste. Le texte qu'on mettra après **<option>** apparaîtra dans la liste. La balise fermante **</option>** est facultative. Les attributs de la balise **<option>** sont :

- **value** : permet d'associer une valeur au choix sélectionné. C'est cette valeur qui est envoyée au serveur.
- **selected** : permet de sélectionner l'élément par défaut. Il n'a pas de valeur particulière.



Les figures ci-contre illustrent une liste déroulante simple et une liste déroulante sur trois lignes avec sélection multiple.



12.3 Formulaires (Complément) :

Il existe d'autres attributs et balises formulaires permettant une facilité d'utilisation et une meilleure mise en forme. Par exemple, l'attribut **tabindex** peut être rajouté à toute balise d'un élément du formulaire. Sa valeur est un entier positif. Il définit l'ordre de sélection de cet élément lorsque la touche *TABULATION* du clavier est utilisée pour parcourir le formulaire.

Par contre, les balises **<fieldset>** et **<legend>** permettent une meilleure présentation du formulaire :

- **<fieldset>** et **</fieldset>** : permet de créer des blocs à l'intérieur du même formulaire.
- **<legend>** et **</legend>** : Pour définir le titre du bloc.