



LE LANGAGE HTML

Génaël VALET

CENTRE DE FORMATION INDIVIDUALISE DU GITA

Lycée Technique Régional Diderot

61, rue David d'Angers - 75019 PARIS

tél. : 01.40.40.36.27/28 — Fax : 01.40.40.36.30

Email : greta.gita@wanadoo.fr

N° SIRET 197 507 122 00046 - code APE 804

SOMMAIRE

SOMMAIRE	2
INTRODUCTION	4
QU'EST CE QUE LE HTML ?	4
UN BREF HISTORIQUE.....	4
LA NORME HTML 4.0	4
LA STRUCTURE D'UN DOCUMENT <i>HTML</i>	5
INTRODUCTION	5
HTML, UN LANGAGE BALISÉ	5
L'ENTÊTE D'UN DOCUMENT HTML.....	5
<i>La balise <HTML></i>	5
<i>La balise <HEAD></i>	5
<i>La balise <TITLE></i>	5
<i>La balise <META></i>	6
La balise META http-equiv	6
<i>La balise LINK</i>	7
LE CORPS D'UN DOCUMENT HTML (BALISE BODY)	7
LES RÈGLES DE SYNTAXE HTML.....	8
LES ACCENTS	8
LE CODE HTML.....	8
LES ÉLÉMENTS HTML	9
LE TITRAGE : ÉLÉMENTS H1,H2,H3,H4, H5 ET H6	9
LA MISE EN FORME DU TEXTE	9
<i>Mise en italique – em ou i</i>	9
<i>Mise en valeur – strong</i>	9
<i>Citation – blockquote ou q</i>	9
<i>Indices et Exposant – SUB ou SUP</i>	10
<i>Paragraphes – P</i>	10
<i>Sauts de ligne – BR</i>	10
LES TABLEAUX – BALISE TABLE.....	10
<i>Nom de tableau – CAPTION</i>	10
<i>Lignes d'un tableau – TR</i>	11
<i>Cellules d'un tableau – TD ou TH</i>	11
<i>Etalement de cellules sur plusieurs colonnes – Attribut colspan</i>	11
<i>Etalement de cellules sur plusieurs lignes – Attribut rowspan</i>	12
LES IMAGES - IMG	12
<i>Images simples</i>	13
<i>Images réactives aux évènements de la souris</i>	13
<i>Images à zones réactives</i>	13
LES HYPERLIENS.....	14
<i>Attribut target</i>	14
<i>Couleurs des liens</i>	14
DIV ET SPAN	14
LES CADRES (FRAMES).....	15
<i>Principe du jeu de cadres</i>	15
<i>Balise FRAMESET</i>	16
Attributs de FRAMESET.....	16
<i>Balise FRAME</i>	17
Liste d'attributs de la balise FRAME.....	17
<i>Conclusion sur les cadres</i>	18
LES OBJETS SCRIPT	18
<i>La balise SCRIPT</i>	18
<i>Évènements intrinsèques</i>	19
LES FORMULAIRES.....	19

<i>La balise FORM</i>	19
<i>Les contrôles de formulaires</i>	20
Attributs des contrôles	20
<i>Elément INPUT</i>	20
Attributs de Input.....	20
<i>L'élément BUTTON</i>	21
<i>Les éléments SELECT et OPTION</i>	21
<i>L'élément TEXTAREA</i>	22
<i>Naviguer dans un formulaire</i>	23
<i>Éléments désactivés ou en lecture seule</i>	23
Éléments désactivés	23
Éléments en lecture seule	24
<i>Soumission de formulaire</i>	24
CONCLUSION	24
LES OUTILS	24

Introduction

Qu'est ce que le HTML ?

Le *HTML* (Hyper Text Markup Langage) est un langage utilisé de façon universelle pour une distribution globale d'informations. Il s'agit d'une langue que tous les ordinateurs en théorie peuvent comprendre. Le *HTML* est le langage utilisé sur le *World Wide Web*.

Le HTML permet de :

- ▶ Publier des documents en ligne contenant des *En-tête*, du *texte*, des *tableaux*, des *listes*, etc...
- ▶ Retrouver des informations en un click de souris grâce aux *Liens Hypertexte*.
- ▶ Concevoir des formulaires permettant de mener à bien des recherches, effectuer des réservations, commander des produits et bien d'autres choses encore.
- ▶ Insérer directement des documents dans d'autres formats, des sources vidéo et sonores et d'autres applications.

Un bref historique

Le *HTML* a connu un nombre important d'évolutions en suivant l'expansion du Web. Il fut créé par *Tim Berners-Lee*. Les spécifications n'étant alors pas définies, l'affichage de pages *HTML* dépendait de l'auteur et du navigateur utilisé. Cet inconvénient majeur a motivé la mise en commun des spécifications pour faire de ce langage un standard.

Plusieurs spécifications du langage firent leur apparition avec le ***HTML 2.0*** , le ***HTML 3.0*** et enfin le ***HTML 3.2*** qui fut un véritable standard grâce aux efforts du groupe de travail sur le *HTML* du consortium sur le *World Wide Web* dès 1996.

HTML a été développé dans le but de fonctionner sur tout type d'ordinateur quel que soit la résolution d'écran, sur des téléphones cellulaires ou sur tout type d'appareil et ceci indépendamment de la bande passante.

La norme HTML 4.0

Cette norme étend les possibilités du *HTML*. Elle permet aux concepteurs de pages d'utiliser des *Feuilles de style (Style sheets)* , des *scripts*, des *cadres (Frames)* , des *objets importés*, du *texte dynamique* et d'améliorer la conception de tableaux et de formulaires.

Aujourd'hui, tous les navigateurs comprennent cette norme et même s'il existe quelques différences d'affichage d'une plate-forme à l'autre, les résultats sont plutôt satisfaisants.

La structure d'un document *HTML*

Introduction

La structure d'un document *HTML* obéit à des règles strictes. Cependant, les navigateurs Web les plus fréquemment utilisés respectent ces règles avec plus ou moins de souplesse. Il en résulte que le concepteur de pages Web devra respecter au maximum ces règles de manière à obtenir un résultat correct quel que soit le navigateur utilisé.

📌 Certains éditeurs *HTML* permettent de générer automatiquement des pages *HTML* grâce à une interface graphique et sans qu'il soit nécessaire de connaître le *HTML*. Cependant, il s'avère que ces logiciels ne sont pas toujours à la hauteur et il vaut mieux connaître le langage pour pouvoir intervenir sur le code *HTML*

HTML, un langage balisé

Le *HTML* est un langage qui permet de décrire un contenu par l'intermédiaire de **balises**. Ces balises encadrent le contenu d'un objet et permettent de déterminer l'endroit exact où commence et où se termine l'objet :

```
<i>Ce texte sera en Italique </i>
<b>Ce texte apparaîtra en gras </b>
<b><i> Ce texte apparaîtra en Gras et en Italique </i></b>
```

L'exemple ci-dessus nous montre ce que les balises sont capables de faire. Toute balise débute par le caractère `<nom_de_la_balise>` et se termine par `</nom_de_la_balise>`

L'entête d'un document HTML

La balise <HTML>

Tout document *HTML* débute par la balise `<HTML>` et se termine par `</HTML>`. Ceci permet de différencier des contenus de diverses natures .

La balise <HEAD>

Cette balise constitue l'entête du document. Elle va contenir une multitude d'informations comme le **type d'encodage des caractères**, les **styles** ou des références aux feuilles de style, le **titre de la page**.

Un document commencera donc par le contenu suivant :

```
<HTML>
<HEAD>
...Contenu de l'entête
</HEAD>
...Reste du document
</HTML>
```

📌 On notera que chaque balise ouvrante précède une balise fermante

La balise <TITLE>

Elle définit le titre du document. Cette balise est obligatoire et doit toujours être contenue dans l'entête du document (Entre `<HEAD>` et `</HEAD>`). Elle s'avère très utile aux moteurs de recherche pour connaître d'emblée le titre de la page.

Notre page HTML se complète :

```
<HTML>
<HEAD>
<TITLE> La page Web du Lycée Diderot </TITLE>
</HEAD>
...Reste du document
</HTML>
```

La balise <META>

Cette balise sert à donner des informations précises sur le document mais se distingue de son contenu. Par exemple, pour spécifier le nom de l'auteur du document, il faudrait écrire le code suivant :

```
<HTML>
<HEAD>
<TITLE> La page Web du Lycée Diderot </TITLE>
<META name="author" content="Genael VALET">
</HEAD>
...Reste du document
</HTML>
```

📌 On remarque que la balise META n'a pas besoin d'être fermée. On peut noter également qu'une balise peut contenir **des attributs**.

Les attributs de cette balise représentent le nom (*name*) et le contenu de la balise (*content*). Le nom de la balise peut être quelconque, cependant les moteurs de recherche utilisent certains mots-clés tels que **KEYWORDS**, **DESCRIPTION** ou encore **REFRESH**. Voici un exemple d'utilisation de la balise META

```
<HTML>
<HEAD>
<TITLE> La page Web du Lycée Diderot </TITLE>
<META name="author" content="Genael VALET">
<META name="description" lang="fr" content="Cette page vous permettra de trouver toutes les
informations nécessaires sur le lycee Diderot">
<META name="keywords" lang="fr" content="lycee, ecole, enseignement, diderot">
<META name="keywords" lang="en-us" content="school, teaching, diderot, high-school">
</HEAD>
...Reste du document
</HTML>
```

📌 On remarque qu'il est possible de spécifier par l'attribut **lang**, la langue utilisée dans la balise. Certains moteurs de recherche utilisent intensivement les balises META pour indexer les pages dans leur base. Il vaut mieux ne pas négliger leur présence (facultative) lorsqu'on conçoit une page Web

La balise META *http-equiv*

Cette balise est **indispensable et obligatoire**. Elle détermine le jeu de caractères que doit utiliser le navigateur pour afficher la page. Ce jeu de caractères répond à des normes ISO (*International Standard Organisation*).

📌 La balise META *http-equiv* est dépendante de la norme ISO. Pour notre alphabet, il convient de choisir la norme ISO 8859-1

Le code d'une page HTML devrait donc commencer par :

```
<HTML>
<HEAD>
  <TITLE> La page Web du Lycée Diderot </TITLE>
  <META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</HEAD>
...Reste du document
</HTML>
```

La balise `META http-equiv` contient un attribut *content* qui signifie que la page sera une page de texte HTML dont le jeu de caractères sera celui de la norme ISO 8859.

La balise `LINK`

Cette balise permet de spécifier des **styles** ou de renvoyer à un lien externe vers une ou plusieurs **feuilles de styles**. Nous reviendrons sur les styles et notamment sur la norme **CSS** (*Cascading Style Sheet*).

Voici un exemple d'utilisation de cette balise :

```
<HTML>
<HEAD>
  <TITLE> La page Web du Lycée Diderot </TITLE>
  <META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <LINK href="mystyles.css" rel="stylesheet" type="text/css">
</HEAD>
...Reste du document
</HTML>
```

La balise `LINK` indique un lien vers un document de type *Feuille de style* qui contient les styles appliqués à la page.

📌 La balise `LINK` peut également servir à indiquer des relations vers d'autres types de documents qui sont en général d'autres documents HTML. Pour plus d'informations, consultez la page Web du W3C : <http://www.w3.org/TR/REC-html40>

Le corps d'un document HTML (Balise `BODY`)

Il s'agit du corps d'un document HTML. Il contient tous les éléments affichés par le navigateur. La balise `BODY` vient juste après l'entête d'un document (Voir page 5).

La balise `BODY` peut posséder des attributs pour définir les couleurs de la page. Voici les principaux attributs de la balise `BODY` :

Attribut	Signification
<code>bgcolor</code>	Couleur de fond de page
<code>Background</code>	Image de fond de page
<code>Text</code>	Couleur du texte
<code>Link</code>	Couleur des liens
<code>Alink</code>	Couleur du lien actif
<code>Vlink</code>	Couleur des liens déjà visités

Attributs de la balise `BODY`

Le document HTML ressemble donc à ceci :

```
<HTML>
<HEAD>
  <TITLE> La page Web du Lycée Diderot </TITLE>
  <META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</HEAD>
<BODY background="white" text="black" link="blue" alink="red" vlink="magenta">
...Corps du document
</BODY>
</HTML>
```

Tous les liens de la page ressembleront à :

[Lien normal](#)

[Lien visité](#)

[Lien actif](#)

📌 La définition des couleurs se fait soit en les appelant par leur nom (black, white, yellow ...) ou encore en donnant leur code RVB en hexadécimal précédé du caractère # (#00FF00 : Rouge à 0 , Vert à 255 et Bleu à 0). Pour coder le bleu, il faudra mettre : `<BODY link="#0000FF" >`

ⓘ Les attributs de la balise *BODY* sont rendus obsolètes par la présence des feuilles de styles dans la norme *HTML 4.0*. Néanmoins, il est possible de les utiliser pour assurer la compatibilité avec les navigateurs de générations plus anciennes (Normes *HTML 3.2*)

Les règles de syntaxe HTML

Les accents

La plupart des plates-formes destinées à visualiser des pages *HTML* étant anglo-saxonnes, la prise en charge des accents n'est pas automatique.

A la rédaction d'une page Web, il ne faut pas écrire les accents directement dans le code.

```
<b>Voici la page du lyc&eacute;e Diderot </b>
```

Le code suivant affichera :

Voici la page du lycée Diderot

Lettres	Code HTML	Lettres	Code HTML
E accent aigu	é ;	A accent circonflexe	â
E accent grave	è ;	A trema	ä ;
E accent circonflexe	ê ;	E trema	ë
A accent grave	à ;	C cédille	ç ;
A accent aigu	á ;		

Codes HTML des principaux accents

Le Code HTML

Le code HTML peut-être tapé au kilomètre sans qu'il soit nécessaire de revenir à la ligne. La mise en forme du texte étant contenu dans les balises, on peut donner libre cours au placement du code. Néanmoins, dans un soucis de lecture confortable du code, il est préférable d'organiser le code de manière arborescente :


```
<HTML>
  <HEAD>
    <TITLE> Lycée Diderot </TITLE>
  </HEAD>
  <BODY>
    <TABLE .....
    </TABLE>
  </BODY>
```

Les éléments HTML

Cette partie va détailler les éléments HTML les plus utilisés dans les pages Web comme les tableaux, le titrage, les liens hypertextes ou encore les images.

Le titrage : éléments H1,H2,H3,H4, H5 et H6

Il est possible de créer des titres afin de faciliter la lecture d'un document et de mettre en valeur certaines parties d'une page Web. Il existe 6 niveaux de titrage.

Le code HTML peut ressembler à ceci :

```
<HTML>
<HEAD>
  <TITLE> La page Web du Lycée Diderot </TITLE>
  <META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <LINK href="mystyles.css" rel="stylesheet" type="text/css">
</HEAD>
<BODY>
  <H1> Les sections du lycée </H1>
  <H2> Les sections BTS </H2>
  <H3> Le BTS &eacute;lectronique </H3>
</BODY>
</HTML>
```

La mise en forme du texte

Mise en italique – em ou i

Balise permettant de mettre en italique

```
<em>Le texte a afficher est en italique</em>
<i> celui la aussi </i>
```

Mise en valeur – strong

Mettre en gras

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

Citation – blockquote ou q

Utilisé pour faire référence à une citation. La plupart des navigateurs affichent un texte indenté.

```
<BLOCKQUOTE cite="http://www.diderot.org/citation.htm" >
  Un coup d'oeil sur l'objet ou sur sa représentation en dit plus long qu'une page de discours
</BLOCKQUOTE>
```

L'attribut CITE est utilisé pour préciser l'origine (Adresse web) d'une citation.

La balise *q* est utilisée pour des citations courtes sans rupture de paragraphes alors que *blockquote* est utilisée pour des citations longues.

Indices et Exposant – SUB ou SUP

Utilisé pour mettre le texte soit en indice, soit en exposant

```
A l'attention de M<SUP>r</SUP> le Proviseur
```

Le texte dans la balise SUP apparaîtra en exposant

Paragraphes – P

Utile pour formater le texte sous forme de paragraphes. Il est possible d'utiliser l'attribut *align* pour centrer ou justifier le texte:

```
<P align="center"> Le paragraphe suivant sera justifié au centre </P>  
<P align="justify"> alors que celui-ci sera doublement justifié</P>
```

Les 4 valeurs possibles pour l'attribut *align* sont *left*, *right*, *center* et *justify*

Sauts de ligne – BR

Les sauts de ligne sont contrôlés par la balise BR. Cette balise n'a pas besoin d'être terminée par une balise *fermante*.

```
La phrase suivante sera terminée par un retour à la ligne <BR>  
Ce qui permettra à la ligne suivante d'être après
```

Les tableaux – balise TABLE

Les tableaux sont très utilisés en HTML car ils permettent une mise en page simple des images et du texte et sont correctement gérés par la plupart des navigateurs.

Les cellules des tableaux peuvent être regroupées et elles peuvent contenir des listes, des paragraphes, des formulaires, des images, du texte préformaté, ou d'autres tableaux.

La balise *TABLE* débute un tableau. Elle peut accueillir les attributs suivants :

Attribut	Signification	Attribut	Signification
align	Spécifie l'alignement du tableau sur la page	cellspacing	Largeur entre les cellules
width	Détermine la largeur du tableau (En % , ou en pixels)	cellpadding	Marge intérieure des cellules
cols	Spécifie le nombre de colonnes du tableau	frame	Spécifie quels côtés de la bordure sont visibles
bgcolor	Couleur de fond des cellules	id	Donne un identifiant au tableau (Utile pour les styles par ex.)
border	Contrôle la largeur de la bordure	class	Détermine le style qui doit être appliqué au tableau

Voici un exemple de code HTML définissant un tableau aligné au centre, large de 75% de la taille de la fenêtre, de couleur de fond noire avec une bordure de 1 point:

```
<TABLE align="center" width="75%" bgcolor="black" border="1" >  
...Code du tableau  
</TABLE>
```

Nom de tableau – CAPTION

CAPTION permet d'ajouter un titre au tableau. Il est possible, grâce à l'attribut *align* de préciser l'endroit où doit se situer le titre (*top*, *bottom*, *left* ou *right*).

❗ La balise *Caption* doit venir immédiatement après la balise de début de tableau *TABLE*. Néanmoins, les navigateurs de dernière génération s'arrangent correctement d'un mauvais placement.*

Lignes d'un tableau – TR

Chaque ligne d'un tableau est considérée comme un ensemble accueillant les cellules. Le tableau suivant contient 3 lignes :

```
<TABLE>
<CAPTION>Tableau &agrave; 3 lignes</CAPTION>
<TR>...Contenu de la ligne 1</TR>
<TR>...Contenu de la ligne 2</TR>
<TR>...Contenu de la ligne 3</TR>
...Reste du tableau
</TABLE>
```

Il est possible d'associer des attributs aux balises *TR*. Ces attributs sont les mêmes que ceux de la balise *TABLE*. L'attribut *align* spécifie l'alignement du texte dans les cellules.

Cellules d'un tableau – TD ou TH

Au sein des lignes, il est possible de définir une ou plusieurs cellules qui peuvent s'étendre sur plusieurs lignes ou plusieurs colonnes.

La balise *TH* est utilisée pour spécifier que la colonne est une colonne de type en-tête. Cela permet aux navigateurs d'afficher de mettre en valeur les en-têtes (Modifiable également par les styles)

Voici un exemple de code HTML d'un tableau et son résultat :

```
<table align="center" border=1 >
<caption>R&eacute;sultats au baccalaur&eacute;at</caption>
<tr align="center" >
  <td>bac sti &eacute;lectronique</td>
  <td>85%</td>
</tr>
<tr align="center" >
  <td>bac sti &eacute;lectrotechnique</td>
  <td>83%</td>
</tr>
</table>
```

Résultats au baccalauréat

bac sti électronique	85%
bac sti électrotechnique	83%

Étalement de cellules sur plusieurs colonnes – Attribut colspan

Chaque ligne étant gérée par une balise *tr*, il suffit d'insérer un attribut *colspan* dans la balise *td* correspondante :

```

<table align="center" border=1 >
<caption>R&eacute;sultats au baccalaur&eacute;at</caption>
<tr align="center" >
  <td>bac sti &eacute;lectronique</td>
  <td>85%</td>
  <td>bac sti &eacute;lectrotechnique</td>
  <td>83%</td>
</tr>
<tr align="center" >
  <td>bac sti m&eacute;canique</td>
  <td colspan="3">83%</td>
</tr>
</table>

```

Voici le r sultat obtenu :

R sultats au baccalaur at

bac sti �lectronique	85%	bac sti m�canique	74%
bac sti �lectrotechnique	83%		

Tableau r sultant de l'utilisation de l'attribut COLSPAN

Etalement de cellules sur plusieurs lignes – Attribut rowspan

Ce qui  tait valable pour les colonnes, l'est pour les lignes gr ce   l'attribut ROWSPAN.

```

<table align="center" border=1 >
<caption>R&eacute;sultats au baccalaur&eacute;at</caption>
<tr align="center" >
  <td>bac sti &eacute;lectronique</td>
  <td>85%</td>
  <td rowspan="2">bac sti &eacute;lectrotechnique</td>
  <td rowspan="2">83%</td>
</tr>
<tr align="center" >
  <td>bac sti m&eacute;canique</td>
  <td >83%</td>
</tr>
</table>

```

R sultats au baccalaur at

bac sti �lectronique	bac sti m�canique	74%
bac sti �lectrotechnique		

Tableau r sultant de l'utilisation de l'attribut ROWSPAN

I Le m lange des attributs ROWSPAN et COLSPAN dans un m me tableau est possible. Cependant, il faut faire attention de ne pas superposer (Chevauchement) les cellules car les r sultats peuvent  tre inattendus.

Les images - IMG

Il est possible d'ins rer des images dans un document *HTML* et de leur donner plusieurs caract ristiques :

- ▶ Images simples.
- ▶ Images r actives aux  v nements de la souris.
- ▶ Images   zones r actives (*Images dont les zones r agissent   la souris et m nent   des liens diff rents*).

Images simples

La balise *IMG* permet de placer une image sur le document :

```
<IMG src="images/diderot.jpg" alt="Logo du lyc&eacute;" >
```

Le code de l'image précédente sera incorporé à l'endroit où l'image doit apparaître. Voici les principaux attributs :

Attribut	Signification	Attribut	Signification
src	URL de l'image à inclure	alt	Texte alternatif (<i>Texte s'affichant en cas de problème avec l'image</i>)
width	Largeur de l'image (<i>facultatif</i>)	align	Alignement de l'image
height	Hauteur de l'image (<i>facultatif</i>)	border	Contrôle la largeur de la bordure (0 par défaut)

Images réactives aux événements de la souris

Ce sont des images qui peuvent réagir et pointer vers un lien. Il suffit pour cela de coupler une balise *A* avec une balise *IMG*.

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

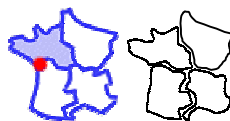
Le code *HTML* précédent implémente une image (*menu4.gif*), qui pointe sur la page *html* *diderot.htm* lorsque l'on clique dessus.

Images à zones réactives

Ces images permettent de créer plusieurs zones de réaction pointant vers des liens différents. Il s'agit de créer une balise *IMG* avec un attribut *usemap* et de créer une autre balise *MAP* qui va contenir les coordonnées des zones réactives et les liens vers lesquelles elles pointent :

```

...
...
<map name="speech">
  <area shape="rect" coords="77,0,178,24" href="sommaire.htm">
  <area shape="circle" coords="35,54,29" href="menu1.htm">
  <area shape="poly"
coords="114,45,186,45,221,41,229,64,192,65,186,93,108,91,112,62,108,45,122,46"
href="menu2.htm">
</map>
```



Exemple d'image réactive à 4 zones représentant les régions

La balise *IMG* pointe vers l'image *speech.gif* en utilisant la carte des zones *my_map*. La carte des zones est représentée par la balise *map* qui contient elle-même les balises *area* représentant la forme et les coordonnées des zones.

Dans notre exemple ci-dessus, il y aura **3 zones réactives** : une rectangulaire, une circulaire et l'autre d'une forme quelconque (polygone).

📌 La balise MAP peut se situer n'importe où dans le document alors que la balise IMG doit figurer à l'endroit où doit apparaître l'image.

Les hyperliens

Ils représentent tout l'intérêt du langage *HTML*. Ils sont la garantie de pouvoir se déplacer sur le *World Wide Web* et de retrouver des informations rapidement. Un lien pointe sur une autre page ou à un endroit précis d'une page.

Il existe 2 balises distinctes pour effectuer des liens. La balise *LINK* et la balise *A*. La balise *LINK* est décrite à la [page 7](#).

La balise **A** reste la plus utilisée et sera bientôt présente en grande quantité sur les pages dont vous aurez la charge.

Un lien se code de la façon suivante :

```
<A href="http://www.diderot.org/sommaire.htm" >Page d'accueil du lyc&eacute;tee Diderot</A>
```

L'attribut *href* désigne l'adresse web de destination du lien. Le texte contenu dans la balise apparaîtra donc comme un lien (*Texte souligné en bleu par défaut*).

📌 Une balise *A* peut contenir autre chose que du texte. Si le contenu est une image, il sera possible de cliquer dessus pour accéder au document pointé par *href*. (Voir le Ch sur les images - P12)

Attribut *target*

Il est possible d'associer à la balise *A* un attribut *target* qui spécifie le cadre de destination du lien. La notion de cadres est abordée plus loin ([Voir P15](#)). Cet attribut va prendre une grande importance pour la cohérence des pages à base de cadres.

```
<A href="http://www.w3.org/" target="central">Organisation W3C</A>
```

Ici, le lien *Organisation W3C* pointe sur l'adresse *http://www.w3.org* et la page sera chargée dans le cadre ayant pour nom *central*.

📌 Il existe des valeurs prédéfinies pour l'attribut *target*. Ces valeurs sont *_blank*, *_parent*, *_top*, *_self*.
_blank : Le document *HTML* pointé par le lien apparaîtra dans une nouvelle fenêtre du navigateur
_parent : Le navigateur chargera le document à la place du jeu de cadre de rang supérieur
_top : Le navigateur chargera le document dans la fenêtre actuelle complète en enlevant tous les cadres
_self : Le navigateur chargera le document dans le même cadre que celui où est placé celui qui l'appelle.

Couleurs des liens

La couleur des liens peut être ajustée grâce aux attributs de la balise *BODY* ([Voir P7](#)) ou par le biais des feuilles de styles. Les feuilles de styles (CSS) offrent des possibilités bien plus étendues pour gérer les couleurs et la mise en page ([Voir module Feuilles de styles - Norme CSS 1](#)).

DIV et SPAN

Les éléments *DIV* et *SPAN*, en conjonction avec les attributs *ID* et *CLASS*, proposent un mécanisme générique pour renforcer la structuration d'un document. Ce sont les deux seuls éléments du *HTML* qui n'ajoutent aucune information de présentation à leur contenu. De ce fait, il est possible de les utiliser associés à des identificateurs et des assignations de classe pour pouvoir formater l'apparence de zones de texte par application ultérieure de feuilles de style.

L'élément *SPAN* est un élément "en ligne" et peut être utilisé à l'intérieur de paragraphes, éléments de liste, etc. lorsque vous souhaitez assigner une déclaration de classe ou des informations de langue à un groupe de mots. L'élément *SPAN* NE PEUT être utilisé pour réunir un groupe d'éléments de niveau "bloc". *SPAN* n'a aucun effet inhérent sur l'apparence tant qu'un style ne lui est pas appliqué, par exemple, via un attribut *STYLE*, ou une feuille de style associée.

L'élément *DIV*, à contrario, est un élément de groupement au niveau "bloc". Il peut être utilisé pour regrouper plusieurs éléments "bloc", mais ne peut pas être utilisé au sein d'un paragraphe. Un

élément *DIV* suivant un paragraphe commencé par un élément *P* et non achevé par sa balise de fin terminera ce paragraphe.

Attribut	Signification	Attribut	Signification
id	Identificateur de bloc	style	Définit un style applicable uniquement à cette balise. Syntaxe de style emprunté à la technologie CSS.
class	Désigne le style à appliquer	lang	Spécifie le code de langue du bloc
align	définit le type d'alignement <i>left</i> , <i>right</i> , ou <i>center</i>		

Attributs de DIV et SPAN

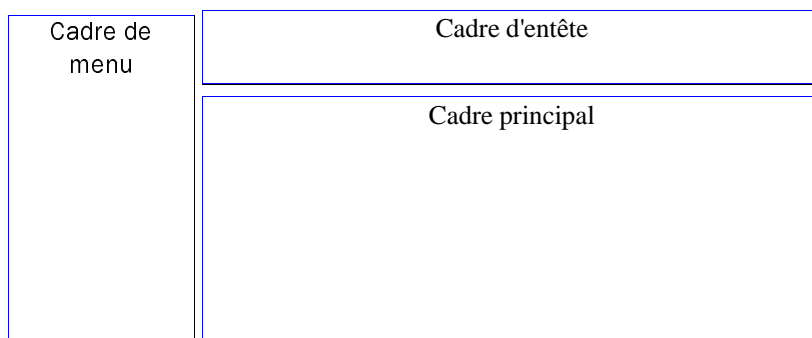
Voici un exemple d'utilisation de la balise *DIV*. Le bloc défini est positionné de manière absolue dans la page grâce à l'attribut *STYLE* dont la syntaxe est conforme à la norme CSS (*Cascading Style Sheet*) :

```
<DIV id="MonBloc" class="MonStyle" style="position:absolute; left:2px; top:20px; width:600px; height:90px"> Bloc de texte MonBloc</DIV>
```

Les Cadres (Frames)

Les cadres *HTML* permettent au concepteur de pages de fractionner les contenus pour aider l'utilisateur à naviguer efficacement. Il devient alors possible de créer des cadres dont le contenu est figé alors qu'un autre cadre est défilant.

Voici un exemple courant d'utilisation des *FRAMES* :



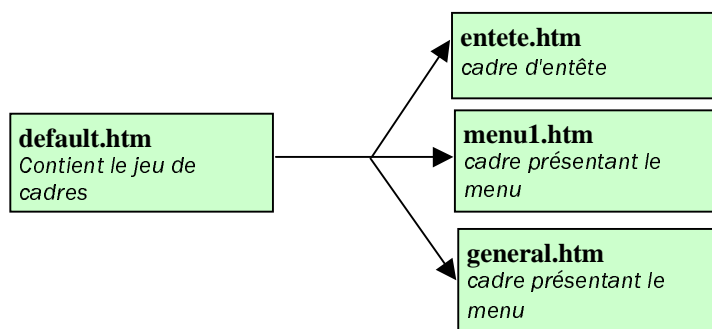
Le cadre de menu contiendra les liens sur les rubriques principales de votre site alors que le cadre d'entête peut contenir une barre de navigation dont le contenu peut dépendre du cadre principal

Principe du jeu de cadres

Un document *HTML* sans jeu de cadres se divise en 2 parties (Voir Chap HEAD P5) : *HEAD* et *BODY*. Une page de cadres contient une partie *FRAMESET*. Cette partie spécifie comment sont disposés les cadres sur la fenêtre du navigateur.

```
<HTML>
  <HEAD>
    ...Contenu de l'entête
  </HEAD>
  <FRAMESET >
    ... Contenu du jeu de cadres
  </FRAMESET>
  <BODY>
    .. Contenu optionnel pour les navigateurs ne gérant pas les frames
  </BODY>
```

Pour créer un jeu de cadres à 3 cadres comme l'exemple ci-dessus, il faudra 4 documents *html* :



Balise FRAMESET

Utilisée pour définir un jeu de cadres, cette balise se situe après l'entête de la page (*HEAD*). Cette balise bénéficie d'attributs permettant de mettre en forme les cadres :

Attribut	Signification
rows	Créer des cadres horizontaux. Il s'agit d'une liste de longueurs séparées par des virgules, s'il n'est pas mentionnés, cet attribut vaut 100%
cols	Créer des cadres verticaux. Il s'agit d'une liste de hauteurs séparées par des virgules, s'il n'est pas mentionnés, cet attribut vaut 100%

Attributs de FRAMESET

Ces deux attributs prennent pour valeur une liste de dimensions séparées par des virgules. Une dimension peut être absolue (un nombre de pixels ou un pourcentage d'écran) ou une dimension relative, indiquée sous la forme "i*", où "i" est un entier.

Lors de l'attribution d'espace aux rangs et aux colonnes, les navigateurs attribuent d'abord les longueurs absolues, puis divisent l'espace qui reste entre toutes les définitions relatives.

```

<FRAMESET rows="50%,50%">
....
</FRAMESET>
  
```

L'exemple précédent spécifie 2 cadres horizontaux de hauteur égale à la moitié de l'écran.

```

<FRAMESET cols="1*,1*,3*">
...le reste de la définition...
</FRAMESET>
  
```

L'exemple précédent construit 3 colonnes. Les chiffres avant les étoiles représentent les fractionnement de l'espace total. Au total, l'espace est divisé en (1+1+3)= 5 parties. La 1^{ère} partie aura une dimension de 1/5 de l'espace, la 2^{ème} de 1/5 et la 3^{ème} de 3/5 de l'espace.

```

<FRAMESET cols="2*,100,1*">
...le reste de la définition
</FRAMESET>
  
```

Dans cet exemple ci-dessus, le navigateur créera un cadre central de 100 Pixels . Si l'écran fait 500 Pixels, il restera alors 400 Pixels à répartir en 2 : 266 Pixels et 133 Pixels.

```

<FRAMESET rows="50*,*50">
...le reste de la définition
</FRAMESET>
  
```

Le navigateur affichera un 1^{er} cadre de 50 pixel de hauteur, un 3^{ème} cadre identique alors que le cadre central prendra le reste.

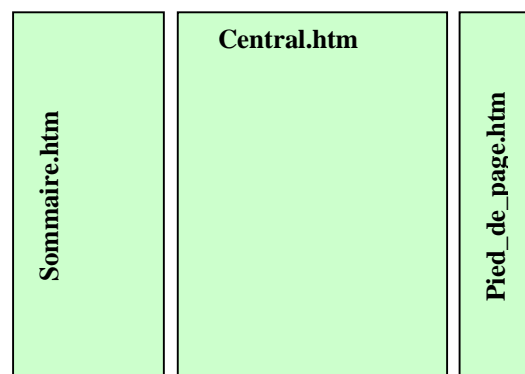
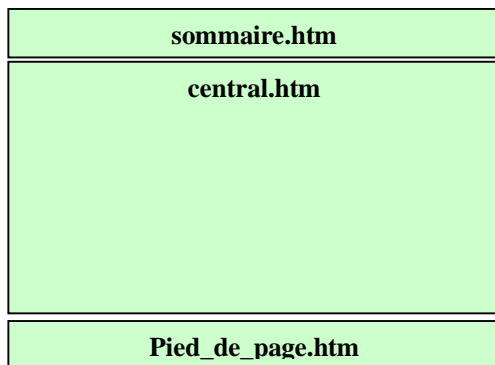
Balise FRAME

Cette balise est utilisée pour décrire préciser le contenu ainsi que les caractéristiques du cadre. Chaque cadre devra contenir une autre page *HTML*.

En reprenant les 2 derniers exemples ci-dessus :

```
<FRAMESET rows="50,*,50"
<FRAME src="sommaire.htm" >
<FRAME src="central.htm" >
<FRAME src="pied_de_page.htm" >
</FRAMESET>
```

```
<FRAMESET cols="2*,100,1*">
<FRAME src="sommaire.htm" >
<FRAME src="central.htm" >
<FRAME src="pied_de_page.htm" >
</FRAMESET>
```




↳ L'intérêt de pouvoir donner des tailles de cadre non figées est que le cadre sera redimensionné automatiquement par le navigateur de manière à pouvoir adapter le contenu qui varie d'une plate-forme à l'autre (Problèmes de résolution et de taille d'écran différentes).

Liste d'attributs de la balise FRAME

Les attributs de cette balise sont importants car ils déterminent les caractéristiques graphiques des cadres tels que les bordures, le contenu, les marges, le scrolling (défilement) et le redimensionnement.

Attribut	Signification	Attribut	Signification
name	Assigne un nom au cadre courant qui pourra être utilisé ultérieurement pour changer le contenu dynamiquement	marginwidth	Cet attribut spécifie l'espace qui doit être laissé sur la gauche entre le contenu du cadre et le bord gauche du cadre lui-même. Cette valeur doit être supérieure à un pixel. La valeur par défaut dépend du navigateur utilisé.
src	Spécifie la localisation du document initialement chargée dans le cadre	marginheight	Cet attribut spécifie l'espace qui doit être laissé en haut entre le contenu du cadre et le bord supérieur du cadre lui-même. Cette valeur doit être supérieure à un pixel.
noresize <i>yes/no</i>	Lorsqu'il existe, avec ses valeurs <i>yes</i> ou <i>no</i> , précise que le volet de fenêtre représentant le cadre ne doit pas être redimensionné	frameborder <i>1/0</i>	Cet attribut indique au navigateur s'il doit y avoir une bordure de cadre ou pas
scrolling <i>yes/auto/no</i>	Cet attribut spécifie la nature et les conditions de présence des barres de défilement du volet.		

Attributs de la balise FRAME

 Il est possible d'imbriquer les jeux de cadres entre eux. C'est à dire, insérer une balise `FRAMESET` au sein d'une autre balise `FRAMESET`

Conclusion sur les cadres

Les pages *HTML* suivantes sont des exemples de jeu de cadres à étudier soigneusement pour bien comprendre le principe des cadres. Ces pages de cadres peuvent se visualiser avec n'importe quel navigateur Web supportant les cadres :

`index1.htm` : Exemple à 3 cadres horizontaux

`index2.htm` : Exemple à 3 cadres verticaux

`index3.htm` : Exemple à jeux de cadres imbriqués.

Les Objets Script

Il s'agit d'objets contenant des petits programmes destinés à s'exécuter sur la machine cliente au moment où le document *HTML* se charge.

Le support des scripts par le *HTML* ne dépend pas du langage utilisé pour ces derniers. Ce qui signifie qu'un navigateur peut supporter plusieurs types de scripts et que cela reste transparent pour le *HTML*.

Les scripts permettent aux concepteurs de pages de proposer une réelle interactivité et du dynamisme en étendant considérablement les possibilités du *HTML* :

- ▶ Écrire dynamiquement une partie ou tout le document au moment de son chargement
- ▶ Travailler conjointement avec les formulaires pour vérifier que la saisie est correcte avant l'envoi. Certains champs d'un formulaire peuvent être remplis automatiquement en fonction de la valeur d'un autre champ
- ▶ S'exécuter en répondant à des événements précis tels que le passage de la souris au dessus d'une image (*OnMouseover*).

Il existe plusieurs types de scripts utilisés aujourd'hui sur Internet :


- ▶ Javascript (Fera l'objet d'un autre document)
- ▶ Vbscript
- ▶ Jscript
- ▶ TCL (Plus rare)

La balise `SCRIPT`

Cette balise marque le début d'un script dans la page *HTML*. Elle peut s'insérer autant de fois que nécessaire dans les balises *HEAD* ou *BODY*. Elle contient plusieurs attributs:

Attribut	Signification
<code>type</code>	Spécifie le langage de script utilisé. Cet attribut doit être un type MIME Internet
<code>langage</code>	Spécifie également le langage mais sous forme d'un identificateur texte simple et non standard
<code>src</code>	Indique la localisation du script lorsqu'il est séparé du document

Attributs de la balise `SCRIPT`

 L'utilisation de scripts sous entend que le navigateur soit à même de prendre en charge le type de langage utilisé. On peut également noter que les navigateurs ont chacun leur manière d'interpréter les scripts.

Exemple d'utilisation de la balise `SCRIPT` :

```
<SCRIPT language="Javascript">  
... Contenu du script  
</SCRIPT>
```

Ou :

```
<SCRIPT type="text/javascript">
... Contenu du script
</SCRIPT>
```

Événements intrinsèques

Ce sont des actions provoquées par un événement particulier. Ce sont des points d'entrée pour un script. Cela signifie qu'un script particulier peut-être exécuté chaque fois que l'événement se produit.

Voici une liste non exhaustive des événements possibles :

Attribut / Signification
<p>onload survient lorsque le navigateur vient de finir le chargement de la fenêtre, ou de tous les cadres. Cet attribut peut être utilisé dans des éléments <i>BODY</i> et <i>FRAMESET</i> (Voir <i>P7</i> et <i>P16</i>)</p>
<p>onunload survient lorsque le navigateur la fenêtre ou un cadre. Cet attribut peut être utilisé dans des éléments <i>BODY</i> et <i>FRAMESET</i> (Voir <i>P7</i> et <i>P16</i>)</p>
<p>onclick , ondblclick survient lors du clic ou double clic de souris sur un l'objet concerné.</p>
<p>onmouseover survient lorsque la souris passe au dessus de l'objet concerné</p>
<p>onmousedown survient lorsque le bouton de la souris est enfoncé mais pas relâché</p>
<p>onmouseup survient lorsque le bouton de la souris est relâché (On suppose qu'il avait été enfoncé)</p>
<p>onmouseout survient lorsque le pointeur de la souris sort de l'objet</p>
<p>onmousemove survient lorsque le pointeur de souris est déplacé sur l'objet concerné</p>

A chaque événement intrinsèque, peut être associé un script qui intervient dès que l'événement se produit pour l'objet spécifié. Ces caractéristiques peuvent donner la possibilité aux concepteurs de pages Web de faire apparaître des messages, des images, de développer une branche de menu et de gérer dynamiquement les événements.

Les formulaires

Un formulaire *HTML* est une section d'un document pouvant contenir des balises et des contrôles. Les contrôles sont des éléments spéciaux qui permettent à l'utilisateur de fournir des données. Les contrôles tels que les cases à cocher, les boutons radio, les labels, les menus, etc. Lorsque l'utilisateur à complété le formulaire, il clique sur un bouton pour envoyer les données qui seront traitées par le serveur.

La balise FORM

Elle contient l'ensemble des balises du formulaire. Tous les contrôles contenus ainsi que les valeurs seront transmis au serveur pour traitement.

Attribut / Signification
<p>action <i>URL</i> précisant l'emplacement du programme chargé de traiter les données ou adresse <i>Mail</i> où envoyer les données</p>
<p>method</p>

Spécifie le type de requête *http* qui sera utilisé pour soumettre les données :
méthode *post* : Données transmises dans le corps du formulaire
méthode *get* : Données transmises dans l'*URL* (Concaténation dans la chaîne *HTTP*)

Le formulaire se présentera comme ceci dans le code *HTML* :

```
<FORM action="http://www.diderot.org/script/chercher.asp" method="post">  
... Contenu du formulaire  
</FORM>
```

ou lorsque le destinataire est une adresse *Mail* :

```
<FORM action="mailto:lycee@diderot.org" method="post">  
... Contenu du formulaire  
</FORM>
```

Les contrôles de formulaires

Les contrôles sont situés le plus souvent dans le formulaire mais il est possible d'utiliser des contrôles en dehors dans les cas où le document *HTML* est utilisé comme une interface utilisateur (*Outils d'administration...*).

Attributs des contrôles

De manière générale, les contrôles sont directement associés à des labels. Cependant, certains contrôles ne suivent pas cette règle. Le label se définit avec l'attribut *label*.

L'attribut *value* d'un contrôle désigne le contenu de ce dernier. Pour un contrôle texte, l'attribut *value* contiendra le texte.

Élément INPUT

L'élément *Type* peut-être de plusieurs types :

- ▶ **text** : champ texte
- ▶ **password** : champ mot de passe
- ▶ **checkbox** : champ case à cocher
- ▶ **radio** : champ bouton radio
- ▶ **submit** : bouton soumettre formulaire
- ▶ **reset** : bouton pour remettre à zéro tous les champs du formulaire
- ▶ **file** : demande à l'utilisateur de désigner un fichier. Lorsque le formulaire est soumis, le contenu de ce fichier sera transmis au serveur comme une valeur de n'importe quel autre contrôle.
- ▶ **hidden** : champ ne devant pas apparaître dans le formulaire
- ▶ **image** : bouton graphique
- ▶ **button** : champ de type bouton

Attributs de Input

Attribut / Signification
type
Indique le type de champ (<i>text, password, checkbox, radio, submit, reset, file, hidden, button</i>)
name
Assigne un nom au contrôle qui est transmis avec le formulaire.
value
Cet attribut spécifie la valeur initiale du contrôle. Il est optionnel sauf lorsque le type de contrôle est "radio", auquel cas il devient obligatoire.

size
Cet attribut indique à l'agent utilisateur la largeur initiale du contrôle. La largeur est donnée en pixels, sauf pour les champs de saisie de texte et de mot de passe pour lesquels elle est exprimée en nombre de caractères (<i>entier</i>).
maxlength
Lorsque le type de champ est "text" ou "password", cet attribut spécifie le nombre maximum de caractères qui peut y être entré. Ce nombre peut être supérieur à la valeur de l'attribut size, auquel cas le navigateur fournira un mécanisme de défilement à l'intérieur du champ. La valeur par défaut correspond à un nombre illimité de caractères.
checked
Lorsque le type de champ est un bouton "radio", cet attribut booléen spécifie que le bouton radio est sélectionné. Cet attribut devra être ignoré pour tous les autres types de contrôles.
src
Lorsque le type de champ est "image", cet attribut spécifie la localisation de l'image utilisée pour "décorer" graphiquement le bouton de soumission.

↳ Tous les attributs concernant les événements intrinsèques (Voir P19) sont pris en charge par ces types de contrôles INPUT. Pour obtenir une description plus détaillée, consultez le document sur la norme HTML donnée en fichier annexe.

L'élément **BUTTON**

Il est identique au type *BUTTON* de l'élément *INPUT* à ceci près qu'il possède des possibilités graphiques plus étendues (*effet de relief*).

L'élément *BUTTON* accepte en son sein d'autre balises de type *IMG* par exemple. Il présente de ce point de vue quelques similitudes avec la balise *A* (Voir **PErreur! Signet non défini**) :

```
<BUTTON type="submit" >
<IMG src="/icons/lycee.gif" alt="Image du lyc&eacute;tee">
</BUTTON>
```

L'exemple précédent insère dans la page, une bouton en forme d'image.

Liste d'attributs :

Attribut / Signification
name
<i>Attribue un nom au contrôle</i>
value
<i>Attribue une valeur</i>
type
<i>button / submit / reset : Caractérise la fonction du bouton. Le formulaire sera soumis si le type est SUBMIT ou réinitialisé si le type est RESET.</i>

*Attribut de la balise **BUTTON***

Vous trouverez un exemple de formulaire dans la page **form1.htm** à visualiser dans votre navigateur.

Les éléments **SELECT** et **OPTION**

L'élément *SELECT* crée une liste d'option (*Menu déroulant*) pouvant être sélectionnée par l'utilisateur. Un élément *SELECT* peut contenir au moins une ou plusieurs entrées *OPTION*.

Voici les attributs de l'élément **SELECT**

Attribut / Signification
name

Cet attribut affecte un nom à l'élément. Ce nom sera appairé avec toute valeur sélectionnée dans la liste, le couple étant alors envoyé au serveur lorsque le formulaire est soumis.

size

Spécifie le nombre de lignes à afficher lorsque le menu est déroulé. Si le nombre de lignes est supérieur à l'attribut SIZE, une gestion de barre de défilement est mise en place.

multiple

Lorsqu'il est présent, cet attribut booléen autorise une sélection multiple. S'il n'est pas mentionné, l'élément SELECT n'autorise qu'une sélection unique. Traditionnellement, les navigateurs affichent des éléments à sélection multiples comme des listes, et les éléments à sélection unique comme des menus déroulant.

Attributs de l'élément SELECT

Chaque ligne d'une liste d'option est spécifié par une balise **OPTION**. Cette dernière possède des attributs :

Attribut / Signification

selected

Cet attribut booléen facultatif spécifie si l'option est sélectionnée ou non au départ.

value

Spécifie la valeur à envoyer lorsque le formulaire est soumis. S'il n'est pas défini, le contenu de l'option est renvoyée

Attributs de l'élément OPTION

Voici un exemple de liste d'option :

```
<SELECT size="4">
  <OPTION value="principal" selected>Sommaire Principal </OPTION>
  <OPTION value="enseign">Sommaire Enseignement </OPTION>
  <OPTION value="bts">Sections BTS </OPTION>
  <OPTION value="bac">Baccalaur&eacute;at</OPTION>
</SELECT>
```

Le même exemple au sein d'un formulaire avec 2 boutons Submit et Reset :

```
<FORM action="mailto:lycee@diderot.org" method="POST">
  <SELECT size="4">
    <OPTION value="principal" selected>Sommaire Principal </OPTION>
    <OPTION value="enseign">Sommaire Enseignement </OPTION>
    <OPTION value="bts">Sections BTS </OPTION>
    <OPTION value="bac">Baccalaur&eacute;at</OPTION>
  </SELECT>
  <INPUT type="submit" value="Send">
  <INPUT type="reset">
</FORM>
```

Il est possible d'utiliser les listes d'options en tant que menus déroulant sur une page HTML. Pour cela, il faut se servir des événements intrinsèques et de Javascript. Le but étant, par exemple, de charger une autre page HTML lorsque l'on clique sur une des entrées de la liste.

L'élément TEXTAREA

Utilisé pour gérer des saisies de texte multilignes, il se différencie de la balise INPUT / Text par la possibilité de gérer plusieurs lignes de saisie.

Attribut / Signification
name
Attribue un nom
rows
Définit le nombre de lignes visibles à l'écran. L'utilisateur pourra utiliser plus de lignes qu'indiqué dans cet attribut mais la gestion de barre de défilement sera mise en place
cols
Spécifie la largeur visible comptée en largeur de caractère moyenne. Les utilisateurs pourront saisir des lignes plus longues que ce qui pourra être vu dans cette largeur, les navigateurs fourniront un mécanisme de défilement du tampon de texte si son contenu déborde de la zone visible. Les navigateurs pourront aussi "casser" les lignes de texte trop longues, de façon à les maintenir visibles dans leur globalité, sans avoir recours au défilement.

Attributs de l'élément TEXTAREA

L'exemple suivant propose un formulaire avec un contrôle de texte multilignes, un bouton de soumission et un bouton Reset :

```
<FORM action="mailto:lycee@diderot.org" method="post">
  <TEXTAREA rows="20" cols="80">
    Premi&egravere ligne du texte initial.
    Deuxi&egraveme ligne du texte initial.
  </TEXTAREA>
  <INPUT type="submit" value="Send"><INPUT type="reset">
</FORM>
```

Il existe d'autres éléments de formulaire tels que FIELDSET, LEGEND, LABEL. Ces éléments sont destinés à améliorer la lisibilité des gros formulaires. Ils sont détaillés dans la norme du W3C disponible en fichier annexe.

Naviguer dans un formulaire

Pour passer d'un contrôle à l'autre dans un formulaire, il est possible d'utiliser la souris ou la touche TAB ou ->|. Dans le cas d'une navigation *tabulée*, il faut définir un ordre logique de navigation. Pour mettre en pratique, il suffit de rajouter un attribut *TABINDEX* dans chaque élément ou contrôle d'un formulaire :

```
<FORM action="mailto:lycee@diderot.org" method="post">
  Nom :<INPUT type="text" value="nom" tabindex=1>
  Pr&eacutenom :<INPUT type="text" value="prenom" tabindex=2>
  <INPUT type="submit" value="Send" tabindex=3><INPUT type="reset" tabindex=4>
</FORM>
```

Éléments désactivés ou en lecture seule

Dans le cas où certains contrôles d'un formulaire ne doivent pas être lus ou modifiés, il existe un attribut permettant de désactiver un élément. Par exemple, un bouton de soumission peut être désactivé tant que les contrôles situés en amont dans le formulaire soient remplis.

Éléments désactivés

L'attribut *disabled* permet de désactiver un élément du formulaire. Désactiver un élément implique que :

- ▶ L'élément ne peut pas être atteint (Pas de "focus")
- ▶ L'élément ne fait plus partie de la liste de tabulation
- ▶ Les valeurs de l'élément ne sont pas transmises lors de la soumission du formulaire

Un élément désactivé apparaît dans la plupart des navigateurs comme grisé.


L'exemple ci-dessous fait apparaître un contrôle désactivé :

```
<INPUT type="text" disabled value="reponse">
```

Éléments en lecture seule

L'attribut *readonly* s'applique aux contrôles ne pouvant être modifiés. Les mêmes restrictions que les éléments désactivés s'appliquent aux éléments en lecture seule (*voir ci-dessus*).

Cependant seuls les éléments *TEXT* et *TEXTAREA* supportent l'attribut lecture seule.

 Ces attributs peuvent être modifiés dynamiquement à l'aide de langage de script tels que le Javascript. (Voir module Javascript)

Soumission de formulaire

Certains contrôles ne sont pas envoyés avec le formulaire :

- ▶ Les contrôles désactivés
- ▶ Les contrôles sans attribut *NAME*

CONCLUSION

Le présent document n'a pas pour vocation de référencer toutes les balises *HTML* ni tous les attributs. Cependant, la norme *HTML 4.0* proposée par le W3C est disponible au format *HTML* et en français dans le fichier *html40-fr.zip* . Il suffit de le décompresser pour avoir accès à la totalité de la référence.

Les outils

La première étape pour apprendre à connaître le *HTML* est d'utiliser un simple éditeur de texte (ex : *bloc-notes de Windows*) ainsi qu'un navigateur.

L'utilisation d'un éditeur *WYSIWYG* (*What you see is what you get*) n'affranchit pas le concepteur de pages *HTML* de connaître le langage. En effet, la plupart de ces concepteurs utilisent leurs connaissances du *HTML* pour parvenir à leurs fins.

En ce qui concerne les outils, retenons :

- ▶ Macromedia Dreamweaver (Outil professionnel modulaire)
- ▶ Microsoft Frontpage (Produit un code *HTML* un peu épais !!)
- ▶ Utilitaire FTP pour télécharger les pages sur un serveur web : CuteFtp ou WsFtp

INDEX

—

_blank · 14
_parent · 14
_self · 14
_top · 14

A

A - balise · 14
accents · 8
action - attribut · 19
align - attribut · 10, 13, 15
alink - attribut · 7
alt - attribut · 13, 15
area - balise · 13
Attributs · 6

B

background · 7
bgcolor · 7, 10
blockquote · 9
BODY - balise · 7
border - attribut · 13, 15
bottom - attribut · 10
br - balise · 10
button - balise · 21
button - input · 20

C

Cadres · 15
CAPTION - balise · 10
cellpadding - attribut · 10
cellspacing - attribut · 10
center - attribut · 10
checkbox - input · 20
checked - attribut · 21
cite - attribut · 9
class - attribut · 10
cols - attribut · 10, 16, 23
colspan - attribut · 11
Couleurs · 8
CSS · 7

D

DESCRIPTION · 6
disabled - attribut · 23
Div · 14

E

em · 9

F

Feuilles de styles · 14
file - input · 20
frame - attribut · 10
FRAME - balise · 17
frameborder - attribut · 17
FRAMESET - balise · 15

G

get - attribut · 20

H

HEAD - balise · 5
height - attribut · 13, 15
hidden - input · 20
HTML - balise · 5

I

i · 9
id - attribut · 10
image - input · 20
IMG - balise · 12
INPUT - balise · 20

J

Javascript · 18
Jscript · 18
justify - attribut · 10

K

KEYWORDS · 6

L

label - attribut · 20
lang · 6
langage - attribut · 18
left - attribut · 10
link - attribut · 7
LINK - balise · 7, 14

M

MAP - balise · 14
marginheight - attribut · 17
marginwidth - attribut · 17
maxlength - attribut · 21
META · 6

method - attribut · 19
multiple - attribut · 22

N

name - attribut · 17, 20, 21, 23
noresize - attribut · 17

O

OnClick - attribut · 19
OnDblclick - attribut · 19
onload - attribut · 19
OnMouseDown - attribut · 19
OnMouseMove - attribut · 19
OnMouseout · 19
OnMouseover - attribut · 18, 19
OnMouseup - attribut · 19
onunload - attribut · 19
OPTION - attribut · 21

P

p - balise · 10
password - input · 20
post - attribut · 20

Q

q · 9

R

radio - input · 20
readonly - attribut · 24
REFRESH · 6
reset - input · 20
right - attribut · 10
rows - attribut · 16, 23
rowspan - attribut · 12

S

SCRIPT - balise · 18
scrolling - attribut · 17
SELECT - balise · 21
selected - attribut · 22
size - attribut · 21, 22
Span · 14
src - attribut · 13, 15, 17, 18, 21
strong · 9
sub · 10
submit - input · 20
sup · 10

T

TABLE - balise · 10
target - attribut · 14
TCL · 18
td - balise · 11
text · 7
text - input · 20
TEXTAREA - balise · 22
th - balise · 11

TITLE - balise · 5
top - attribut · 10
tr - balise · 11
type - attribut · 18, 20, 21

U

usemap - attribut · 13

V

value - attribut · 20, 21, 22
Vbscript · 18
vlink - attribut · 7

W

width - attribut · 10, 13, 15