

Premiers pas en HTML

par *Emmanuel Fleury*

\$Date: 1999/09/20 07:50:31 \$

Ce document présente les bases du langage HTML. Les règles de syntaxe, les règles de création d'un document, les principales balises et leurs paramètres, les balises avancées (liens, images, listes et tableaux).

Table des matières

1	Introduction	2
1.1	À propos de cette documentation	2
1.2	Qu'est-ce que HTML ?	2
2	Comment ça marche ?	3
2.1	Une vague idée du protocole HTTP	3
2.2	Le moteur de rendu HTML	3
2.3	Les serveurs Web	3
2.4	Comment faire sa page ?	4
2.5	Ce qu'il faut avant de commencer	4
3	La syntaxe de base	4
3.1	Les langages à balise	4
3.2	Les règles de base	5
4	La structure d'un document	5
4.1	L'entête	5
4.1.1	<TITLE>	6
4.1.2	<META>	6
4.2	Le corps	6
5	Les principales balises	7
5.1	La mise en page	7
5.2	Les styles	8
5.3	Les environnements	8
6	Les balises avancées	9
6.1	Les liens hypertextes	9
6.1.1	Les URL	9
6.1.2	L'adresse d'un autre site	10
6.1.3	Les adresses de fichiers	10

6.1.4	Une adresse dans la page	10
6.2	Les images	11
6.3	Les listes	11
6.3.1	Les listes simples	11
6.3.2	Les listes ordonnées	11
6.3.3	Les listes de définitions	12
6.4	Les tableaux	12
7	Références	14
8	Copyright	14

1 Introduction

1.1 À propos de cette documentation

Cette documentation n'a pas vocation à être exhaustive. Son but est de donner à ceux qui débutent suffisamment d'indications pour être indépendants en HTML. Une liste de sites Web et de livres est placée à la fin de ce document pour plus ample information.

1.2 Qu'est-ce que HTML ?

Le langage HTML (Hyper Text Markup Language) est un langage de formatage de documents hypertextes. Autrement dit, il sert à définir la structure et le contenu des pages Web.

Le langage en lui-même fut développé en 1990 par Tim Berners-Lee pendant sa thèse au CERN (Genève). Il fut rapidement popularisé par Mosaic (le premier navigateur Web). Dès la version 2.0, les spécifications de ce langage ont été confiées à un groupe international indépendant, le « *World Wide Web Consortium* » (W3C). Ce consortium rassemble des universités, des entreprises et des organismes publics de différents pays (DARPA, MIT, INRIA, ...).

L'indépendance du W3C a permis de mettre sur un pied d'égalité toutes les entreprises et les utilisateurs du *standard* HTML. Cette structure permet d'éviter toute prise de monopole par l'un des acteurs du marché (comme c'est souvent le cas en informatique). Ainsi, les sociétés rivalisent pour essayer de fournir le navigateur le plus compatible possible avec la norme HTML en préservant le standard commun.

Le langage HTML lui-même est spécifié en *SGML* (Standard Generalized Markup Language). Ce langage permet de décrire d'autres langages du type de HTML. SGML est compliqué, mais il a l'avantage de pouvoir être traduit facilement en plusieurs formats (HTML, Postscript, ASCII, RTF). Cela permet de n'écrire qu'une seule fois le document et ne pas avoir à le retoucher suivant le format désiré (page Web, document papier, lecture sur ordinateur de poche, ...).

Cependant, au fil du temps, SGML s'est avéré très limité sur certains points (complexité d'utilisation, liens trop étroits entre le contenu d'un document et sa présentation, faiblesse du pouvoir d'expression, manque de flexibilité, ...). Le W3C a donc commencé à travailler sur un nouveau standard. Ce langage, encore en développement à l'heure actuelle, a été appelé *XML* (eXtensible Markup Language). Un langage HTML basé sur XML est aussi en cours de développement (*XHTML*).

Mais ceci est une autre histoire...

Ce document traite du *HTML 4.01*. On notera les expressions HTML : `<BLINK>Kitch!</BLINK>`

2 Comment ça marche ?

2.1 Une vague idée du protocole HTTP

Pour faire simple, lorsque vous regardez une page Web sur Internet, cela se passe comme ça :

1. Lorsque vous demandez à votre navigateur Web d'afficher une page, en tapant son adresse : `http://www.w3.org/TR/REC-html40/about.html`.
2. Votre navigateur essaye de contacter, via l'Internet, l'ordinateur nommé : `www.w3.org`.
3. Sur `www.w3.org`, un logiciel veille, c'est un *serveur Web*. Dès qu'il repère la demande de votre navigateur, il envoie une confirmation qui précise qu'il est bien vivant et qu'il est prêt à traiter une demande.
4. Votre navigateur envoie une demande pour le fichier `/TR/REC-html40/about.html`.
5. Le serveur Web envoie le fichier demandé à votre navigateur.
6. Une fois ce fichier reçu, votre navigateur utilise le programme HTML du fichier `about.html` pour construire le document.
7. Le navigateur Web affiche le résultat sur votre écran.

2.2 Le moteur de rendu HTML

Il est important de comprendre que chaque navigateur possède sa propre interprétation du langage HTML. La transformation du document HTML en la page que vous voyez sur votre écran est le fait d'un sous-programme appelé *moteur de rendu HTML* qui est propre au navigateur. Chaque moteur étant, selon les cas, plus ou moins à jour sur les normes du W3C, ou plus ou moins bogué. Par exemple, il existe des pages qui ne donneront pas la même chose selon que vous la regarderez avec Netscape ou avec Internet Explorer.

2.3 Les serveurs Web

Un serveur Web est le logiciel qui permet aux utilisateurs de consulter des pages Web. Il en existe un grand nombre. Cependant, les plus connus sont *Apache* (logiciel libre), *IIS* (Microsoft) et *Netscape-Entreprise* (Netscape). Le serveur Web le plus utilisé est Apache (cf *Netcraft* `<http://www.netcraft.com/survey/>`).

Le service de base d'un serveur Web est d'envoyer les fichiers qui décrivent des pages Web aux utilisateurs qui en font la requête.

Mais il peut faire bien mieux que cela !

L'exemple le plus flagrant sont les moteurs de recherches consultables sur le Web (Infoseek, AltaVista, Google, ...). Il est absolument impossible de stocker, sous forme de pages HTML, **toutes** les requêtes possibles des utilisateurs.

En fait, un petit programme va servir d'interface entre l'utilisateur et un logiciel de base de données. Ce programme va traduire votre requête pour interroger la base de donnée et insérer les résultats obtenus dans un squelette prédéfini de page Web. Les pages sont donc construites *à la volée* et non pas stockées sur le serveur.

2.4 Comment faire sa page ?

Pour cela, il vous faut un ordinateur et un éditeur de texte. Personnellement, je conseille Emacs ou XEmacs, mais le Notepad de Windows peut tout aussi bien faire l'affaire (bien qu'il ne soit pas aussi évolué).

Évidemment, il existe des tas de logiciels très sophistiqués qui permettent de faire des pages HTML sans toucher au langage HTML, mais ils génèrent rarement du code propre (souvent lourd, redondant, ou même inutile). Qui plus est, vos possibilités sont restreintes aux capacités du logiciel qui offre toujours moins de possibilité que le langage HTML lui-même.

2.5 Ce qu'il faut avant de commencer

Avant de vous lancer dans la conception d'une page Web, il faut deux choses :

1. Avoir un ordinateur relié à l'Internet (de préférence en permanence).
2. Avoir un logiciel de serveur Web qui fonctionne correctement.

Remplir ces deux conditions demandent à la fois un matériel encore inabordable pour un particulier et des connaissances techniques importantes.

Il existe cependant un grand nombre de *fournisseurs d'accès à l'Internet* (FAI) qui vous offrent un peu d'espace disque sur une machine ayant accès à l'Internet et les compétences de leurs administrateurs système. Ces organismes peuvent être des entreprises privées ou des organismes publics (universités, centre de recherche, etc).

Faites attention à bien respecter la charte que votre prestataire vous présentera lors de la création de votre site. Elle vous semblera sûrement restrictive et aliénante, mais sachez qu'elle est surtout là pour éviter à votre prestataire de tourner court à cause des irrégularités de l'un de ses client (cf *L'affaire Estelle Halliday* contre Altern.org).

Enfin, une fois le site prêt, votre prestataire vous communiquera tous les détails techniques nécessaires à son installation. Comme ils diffèrent tous je n'en parlerai pas ici.

3 La syntaxe de base

La syntaxe du langage HTML est très simple. Il fait parti d'une famille de langage que l'on appelle communément *langages à balises*. La syntaxe de ce langage peut se résumer en six règles très simples.

3.1 Les langages à balise

HTML est avant tout un langage à balises. C'est à dire un langage qui permet de définir des zones qui ont certaines propriétés.

Par exemple, si je veux écrire en gras, je vais définir une zone où tout sera en gras :

```
<GRAS>Cette phrase est en gras</GRAS>
```

Ce qui donnera :

Cette phrase est en gras

Le langage HTML est à peine plus compliqué (voir plus loin). Dès lors, la principale difficulté du HTML revient à connaître et savoir correctement utiliser les différentes balises.

3.2 Les règles de base

Voici les cinq règles de base pour faire du HTML :

1. Définir une zone : `<BALISE>Zone</BALISE>`

Lorsqu'on définit une zone, on suppose que tous les objets qui sont à l'intérieur possèdent une certaine propriété (par exemple ils sont écrits en gras).

2. Balise ponctuelle : `<BALISE>`

Certaines balises ne nécessitent pas d'être refermées car elles ne définissent pas de zone à proprement parler, mais un endroit où une action se produit (par exemple la balise `<BR>` saute à la ligne).

3. Paramétrage d'une balise : `<BALISE parametre="valeur">`

Il existe toujours un grand nombre de paramètres pour une balise (par exemple dans la balise `<P>`, qui marque les paragraphes, on peut préciser le paramètre `align = left|center|right|justify` qui permet de placer le paragraphe en question, respectivement, à gauche, au centre, à droite ou de le justifier). Il est important de savoir que certains paramètres sont obligatoires (par exemple dans la balise `<IMG src="Pelisse.png">` affiche l'image `Pelisse.png`).

4. Imbrication des balises : `<BAL1>...<BAL2>...</BAL2>...</BAL1>`

C'est le principe des poupées russes. Croiser les balises provoque une erreur qui est souvent interprétée de façon aléatoire par les navigateurs.

5. Commentaires : `<!-- Ceci est un commentaire -->`

Il est possible de sauter des lignes dans un commentaire.

4 La structure d'un document

Nous allons à présent passer à la conception d'une page Web. Nous commencerons par décrire la structure d'un document HTML.

On délimite un document HTML par la balise `<HTML>`. Ce document contient deux parties essentielles :

- L'entête du document (balise `<HEAD>`),
- Le corps du document (balise `<BODY>`)

Par exemple :

```
<HTML>
  <HEAD>
    <TITLE>Mon premier document HTML</TITLE>
  </HEAD>

  <BODY>
    <P>Salut tout le monde!!
  </BODY>
</HTML>
```

4.1 L'entête

L'entête sert à donner des informations qui n'apparaissent pas sur le document. Cela peut-être le titre de la fenêtre dans laquelle est affichée la page Web (`<TITLE>`), le nom de l'auteur ou bien des mots clefs décrivant la page pour les moteurs de recherche (`<META>`),

4.1.1 <TITLE>

Ce champ sert à donner un titre cohérent à votre page (autre qu'un nom de fichier). Ce champ sert notamment à nommer la fenêtre du navigateur dans laquelle apparaît la page ou encore à donner un nom au signet qui est créé lorsqu'on marque votre page. Il vaut mieux donner un titre court et explicite.

4.1.2 <META>

Le champs <META> regroupe des informations sur votre document mais qui n'y apparaît pas. Cela regroupe toutes informations utiles aux moteurs de recherches, aux navigateurs ou aux administrateurs du site.

C'est une balise ponctuelle ; les informations qu'elle véhicule sont passées en paramètres. Elle s'utilise comme suit : <META name="nom" lang="langue" content="info">

Par exemple :

```
<HTML>
  <HEAD>
    <META name="Author" content="Jean Dupont">
    <META name="Date" content="1994-11-06T08:49:37+00:00">
    <META name="keywords" lang="en-us"
      content="vacation, Japan, Osaka">
    <META name="keywords" lang="en"
      content="holiday, Japan, Osaka">
    <META name="keywords" lang="fr"
      content="vacances, Japon, Osaka">

    <TITLE>Mes vacances au Japon</TITLE>
  </HEAD>

  <BODY>
    <P>Le Japon est un beau pays ...
  </BODY>
</HTML>
```

Lorsqu'un navigateur lira cette page, il connaîtra son auteur et la date de création. Il pourra les communiquer à la demande de son utilisateur.

De même, lorsqu'un moteur de recherche voudra cataloguer cette page, il saura immédiatement que les mots clefs, fixé par l'auteur lui même, sont selon les langues utilisées : vacation, holiday, Japan, Osaka, vacances, Japon.

Les champs META sont loin d'être standards et, donc, sont tout à fait optionnels, mais cela permet souvent aux navigateurs de disposer de plus d'informations et aux moteurs de recherche de mieux cibler votre page.

4.2 Le corps

Le corps du document HTML sert à définir la zone qui contient la description du document visualisé. C'est ici que l'on définit le contenu et la présentation du document. Plusieurs paramètres peuvent être fixés :

- **background** : Ce paramètre permet de donner l'adresse d'une image qui servira de fond à votre page.
Exemple : <BODY background="/pictures/primavera.png"> ,
- **text** : Permet de fixer la couleur du texte pour tout le document,
- **link** : Permet de fixer la couleur des liens non visités,

- `vlink` : Permet de fixer la couleur des liens visités,
- `alink` : Permet de fixer la couleur des liens sélectionnés par l'utilisateur,
- `bgcolor` : Permet de fixer la couleur du fond de la page.

Quelques couleurs :

`black = "#000000"`, `green = "#008000"`, `silver = "#C0C0C0"`, `lime = "#00FF00"`, `gray = "#808080"`,
`olive = "#808000"`, `white = "#FFFFFF"`, `yellow = "#FFFF00"`, `maroon = "#800000"`, `navy = "#000080"`,
`red = "#FF0000"`, `blue = "#0000FF"`, `purple = "#800080"`, `teal = "#008080"`, `fuchsia = "#FF00FF"`,
`aqua = "#00FFFF"`

Une couleur qui n'est pas dans cette liste s'écrit toujours en code RGB et en hexadécimal sous la forme :
"`#8FA565`". "8F" représentant l'intensité du rouge, "A5" l'intensité du vert et "65" l'intensité du bleu.

Par exemple :

```
<HTML>
  <HEAD>
    <TITLE>Une leçon de mauvais goût!</TITLE>
  </HEAD>
  <BODY bgcolor="cyan" text="red"
    link="red" alink="#FF00FF" vlink="maroon">
    ... corps du document...
  </BODY>
</HTML>
```

5 Les principales balises

Passons, à présent, au contenu du document.

5.1 La mise en page

La mise en page regroupe tout ce qui sert à formater le document. À savoir les sauts à la ligne, les paragraphes, etc.

- `<BR>` :
Cette balise ponctuelle (on a pas besoin de la fermer) sert à forcer le saut à la ligne (`BR`eak). La plupart des moteurs de rendu HTML ajustent la longueur des lignes en fonction de la largeur de la fenêtre dans laquelle on visualise la page HTML. Mais il arrive parfois que l'on veuille forcer un saut de ligne.
- `<P>` :
Cette balise sert à notifier un début de paragraphe (`Parag`raph). On peut y préciser le paramètre `align = left|center|right|justify` qui permet, respectivement, de positionner le paragraphe aligné à gauche, centré, aligné à droite ou justifié.
- `<H1>` à `<H6>` :
Ces balises permettent de faire apparaître une certaine hiérarchie dans les titres de documents. Le `H1` sert aux gros titres, le `H3` est la taille de la police standard et le `H6` est la plus petite. On peut y préciser le paramètre `align`.
- `<HR>` :
Cette balise ponctuelle permet d'afficher une ligne de séparation (`Horiz`ontal `R`ule). On peut fixer les paramètres :
 - `align = left|center|right` Permet d'aligner la ligne de séparation à gauche, au centre ou à droite.
 - `noshade` : Est un paramètre booléen qui enlève l'ombrage de la barre s'il est présent.

- **size** : Ajuste la taille de la ligne de séparation. On peut avoir deux unités de mesure. L'une absolue, en *pixel*, il suffit de ne mettre qu'un chiffre (`size="49"`). L'autre relative à la largeur de la fenêtre du navigateur qui visualise la page, en *pourcent* (`size="50%"`).
 - **width** : Ajuste la hauteur de la ligne de séparation. On peut aussi utiliser des pixels ou des pourcentages.
- Quelques exemples de barre de séparation :

```
<HR width="50" align="center">
<HR size="5" width="50%" align="center">
<HR noshade size="5" width="50%" align="center">
```

5.2 Les styles

Le style est ce qui permet de faire ressortir certains mots du texte, en le mettant en *italique* ou bien en **gras** par exemple.

- **** : Est la balise qui permet de mettre en *valeur* une partie du texte (**EM**phase),
- **** : Permet de mettre une partie du texte en **gras** (**B**old),
- **<IT>** : *Italique*,
- **<TT>** : Permet d'écrire en caractères machine à écrire (**TT**ypewriter Text),
- **<STRIKE>** : Caractères barrés,
- **<BLINK>** : Permet de faire clignoter la partie du texte concernée,
- **<BIG>** : Écris en grand,
- **<SMALL>** : Écris en petit,
- **<Q>** : Citation (Quotation),
- **<SUP>** : Exposant,
- **<SUB>** : Indice,
- **** : Agrandit ou réduit le texte concerné.

5.3 Les environnements

Les environnements permettent de formater le texte d'une certaine manière. Par exemple pour faire des citations ou pour afficher le listing d'un programme sur la page.

- **<CODE>** : Permet d'afficher du code,
Par exemple :

```
<CODE>
#include <stdio.h>

void main() {
    printf("Hello World!\n");
}
<\CODE>
```

Donne :

```
#include <stdio.h>

void main() {
    printf("Hello World!\n");
}
```

- **<PRE>** : Permet conserver le formatage du texte tel qu'il est (**PRE**formatted),
- **<BLOCKQUOTE>** : Permet de faire une citation dans un bloc de texte.

6 Les balises avancées

Les balises avancées sont celles qui ont, au moins, un paramètre obligatoire ou qui ont une structure plus complexe.

6.1 Les liens hypertextes

Les liens hypertextes permettent de relier une partie du texte à un autre endroit du Web. Cela peut être un autre endroit sur cette même page, un autre endroit sur le même site, ou bien un autre endroit sur un autre site. On fait référence à cet autre endroit grâce à une adresse Web (URL, Uniform Resource Locators).

Un lien hypertexte se présente ainsi : `<A href="URL">Un texte</A>`

Le A fait référence à Anchor (ancree).

6.1.1 Les URL

Les URL sont de la forme :

`service://serveur/chemin/fichier`

Les services proposés :

- `http` : Référence un serveur Web (c'est le plus fréquent),
- `ftp` : Référence un serveur de fichiers,
- `news` : Référence un serveur de forum de discussions,
- `mailto` : Référence une adresse *e-mail*,
- `file` : Référence un fichier sur votre ordinateur.

Le nom du serveur : Le nom du serveur est composé de plusieurs parties. Généralement, la première partie identifie le service qu'il rend (`www` ou `ftp`). Puis, vient une localisation plus précise du serveur (nom d'université, nom d'entreprise, etc). Enfin, vient un identificateur qui permet de savoir à quel type d'organisme on a affaire et/ou dans quel pays il se trouve :

- `org` : Organisation non gouvernementale,
- `com` : Entreprises commerciales,
- `gov` : Gouvernement américain,
- `edu` : Éducation,
- `mil` : Militaire,
- `net` : Organismes fournisseurs d'Internet,
- `fr` : France,
- `gouv.fr` : Gouvernement Français,
- `be` : Belgique,
- `de` : Allemagne,
- `uk` : United Kingdom (Royaume-Unis),
- `ch` : Confédération Helvétique (Suisse),
- `ca` : Canada,
- Et bien d'autres...

Chemin et nom du fichier : Le chemin donne la position exacte du fichier dans l'arborescence de fichiers de la machine qui contient le site Web et le nom du fichier qui va être lu.

Un exemple : Prenons : `http://www.w3.org/TR/REC-html40/about.html`

On voit que le service sera un serveur Web (`http`), qui se nomme `www.w3.org` et que le chemin jusqu'au fichier `about.html` est `/TR/REC-html40/`.

6.1.2 L'adresse d'un autre site

Pour pointer vers un autre site que le siens, il suffit placer son URL dans la balise.

Par exemple :

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

Donne : *W3C*

6.1.3 Les adresses de fichiers

On peut aussi pointer sur un fichier qui est présent sur l'ordinateur qui accueille le site (cela sert à relier les pages du site entre-elles).

L'une des solutions est de faire systématiquement une référence à l'adresse complète du site, mais cela devient vite pénible. L'autre solution, est de donner le chemin vers le fichier par rapport à la position du fichier actuellement lu.

Par exemple, si vous êtes en train de consulter le fichier `/music/toc.html` et que vous voulez faire un lien vers `/music/tchaikovsky/biography.html`, il suffit de faire appel à `tchaikovsky/biography.html`.

6.1.4 Une adresse dans la page

On peut aussi faire référence à un point précis dans la page, par exemple dans une table des matières ou pour revenir en haut d'une page très longue. Pour cela, il faut avoir défini des points d'ancrage sur la page.

Un point d'ancrage se définit ainsi : `<A NAME="label">Point d'ancrage</A>`

Et la référence à ce point d'ancrage ainsi : `<A HREF="#label">Référence à mon point d'ancrage</A>`

Par exemple :

```
<H1 align=center>Alice's Adventures under Ground</H1>
<H4 align=center>By Lewis Carroll</H4>
<P><A href="#chapter1">Chapter 1</A><BR>
<A href="#chapter2">Chapter 2</A><BR>
<A href="#chapter3">Chapter 3</A><BR>
...
<H2><A name="chapter1">Chapter 1</A></H2>
Alice was beginning to get very tired of sitting ...
<H2><A name="chapter2">Chapter 2</A></H2>
They were indeed a curious looking party that assembled ...
<H2><A name="section2.1">Chapter 3</A></H2>
"The first thing I've got to do," said Alice to herself, ...
```

6.2 Les images

On peut inclure facilement une image dans un document HTML grâce à la balise `IMG`. On l'utilise ainsi : `<IMG SRC="nom du fichier">`.

Les paramètres que l'on peut modifier sont :

- `alt` : Permet de donner une alternative à l'image lorsque le navigateur Web ne peut afficher l'image,
- `align` : Options d'alignement `align=left|center|right|justify` (comme précédemment),
- `width` : Largeur de l'image,
- `height` : Hauteur de l'image,
- `border` : Épaisseur du bord de l'image,
- `vspace` : Espace vertical entre les bords de l'image et un autre objet (texte, autre image, etc),
- `hspace` : Espace horizontal entre les bords de l'image et un autre objet (texte, autre image, etc),

Par exemple :

```
<IMG src="Botticelli12.png" alt="Venus sortant des eaux" align=left border="0">
```

6.3 Les listes

Il existe trois types de listes en HTML, les listes ordonnées (`OL`, `Ordered List`), les listes non ordonnées (`UL`, `Unordered List`) et les listes de définitions (`DL`, `Definition List`).

En HTML, une liste est composée d'une balise qui indique que l'on est dans un environnement liste et d'une balise ponctuelle qui marque le début de chaque item.

Par exemple :

```
<OL>
  <ITEM>Premier item de la liste.
  <ITEM>Deuxième item de la liste.
    <UL>
      <ITEM>Premier item de la liste du deuxième item!
    </UL>
</OL>
```

Donne :

1. Premier item de la liste.
2. Deuxième item de la liste.
 - Premier item de la liste du deuxième item !

6.3.1 Les listes simples

On la marque avec la balise `<UL>` (`Unordered List`). Les items sont marqués par la balise ponctuelle `<LI>`. Les paramètres sont :

- `type=disc|circle|square` : Qui permet de fixer le symbole qui apparaît devant les items,
- `compact` : Qui permet de tasser un peu la liste.

6.3.2 Les listes ordonnées

On la marque avec la balise `<OL>` (`Ordered List`). Les items sont marqués par la balise ponctuelle `<LI>`. Les paramètres sont :

- `type=1|a|A|i|I` : Qui permet de fixer le symbole qui apparaît devant les items. 1 pour une numérotation classique (1, 2, 3, ...), a ou A pour une numérotation alphabétique (a, b, ...), i ou I pour une numérotation latine (i, ii, iii, ...),
- `compact` : Qui permet de tasser un peu la liste.

Par exemple :

```
<OL type=1 compact>
  <LI> texte 1
  <LI> texte 2
</OL>
```

Donne :

1. texte 1
2. texte 2

6.3.3 Les listes de définitions

On la marque avec la balise `<DL>` (**D**efinition **L**ist). Il y a deux items possibles, le terme à définir : `<DT>` et la définition en elle-même : `<DD>`.

Par exemple :

```
<DL>
  <DT> Terme 1 à définir
    <DD> Définition du terme 1
  <DT> Terme 2 à définir
    <DD> Définition du terme 2
</DL>
```

Donne :

Terme 1 à définir

Définition du terme 1

Terme 2 à définir

Définition du terme 2

6.4 Les tableaux

Les tableaux sont des structures importantes en HTML. Elles servent non seulement à faire des tableaux comme on a l'habitude d'en voir, mais aussi à gérer la mise des documents.

Vous voulez mettre une marge à votre document ? Vous voulez un agencement particulier de différentes images ? *Utilisez les tableaux !*

Les différentes balises qui permettent de construire un tableau sont :

- `<TABLE>` Définis la zone qui contiendra les objets qui formeront le tableau,
- `<CAPTION>` Définis la légende du tableau,
- `<TR>` (**T**able **R**ow) Définis une ligne du tableau,
- `<TD>` (**T**able **D**ata **c**ell) Définis une cellule du tableau,
- `<TH>` (**T**able **H**ead**e**r) Définis les entêtes du tableau. C'est-à-dire la première ligne ou la première colonne.

Par défaut, l'alignement est centré et la police de caractères est en gras.

Certains navigateurs laxistes permettent l'omission des balises de fin de <TR>, <TD> et <TH>, mais c'est souvent une source d'erreurs et/ou d'ambiguïtés. Il vaut donc mieux refermer ces balises.

Par exemple :

```
<TABLE border="1">
<CAPTION>Ceci est un exemple</CAPTION>
  <TR>
    <TD align=center>Première case</TD>
    <TD align=center>Deuxième case</TD>
  </TR>
  <TR>
    <TD align=center>Troisième case</TD>
    <TD align=center>Quatrième case</TD>
  </TR>
</TABLE>
```

Donne :

Ceci est un exemple	

Première case	Deuxième case

Troisième case	Quatrième case

Les détails de quelques paramètres :

- TABLE :
 - summary : Résumé de ce que contient la table.
 - width : Largeur du tableau, exprimé en pixel ou en pourcentage.
 - align=left|center|right|justify Position du texte dans la cellule.
 - border : Épaisseur du trait en pixel.
 - frame=void|above|below|hsides|lhs|rhs|vsides|box|border : Détermine quels bords seront visibles.
 - void : aucun bord visible,
 - above : le haut uniquement,
 - below : le bas uniquement,
 - hside : le haut et le bas,
 - lhs : le côté gauche,
 - rhs : le côté droit,
 - vsides : le côté gauche et le côté droit,
 - border : les quatre bords.
 - cellpadding : Espace entre les cellules.
- TR :
 - align=left|center|right|justify : Position du texte dans la cellule.
 - valign=top|middle|bottom : Alignement vertical au sein de la cellule.
 - top : Aligné en haut,
 - middle : Au centre,
 - bottom : Aligné en bas.
- TH :
 - align=left|center|right|justify : Position du texte dans la cellule.
 - valign=top|middle|bottom : Alignement vertical.
 - colspan : Permet de recouvrir plusieurs colonnes avec une seule cellule.

- nowrap : Empêche le passage à la ligne dans une cellule.

Un exemple plus compliqué :

```
<TABLE border="1"
  summary="This table gives some statistics about fruit
  flies: average height and weight, and percentage
  with red eyes (for both males and females).">
<CAPTION><EM>A test table with merged cells</EM></CAPTION>
<TR><TH rowspan="2"><TH colspan="2">Average
  <TH rowspan="2">Red<BR>eyes
<TR><TH>height<TH>weight
<TR><TH>Males<TD>1.9<TD>0.003<TD>40%
<TR><TH>Females<TD>1.7<TD>0.002<TD>43%
</TABLE>
```

Donne :

A test table with merged cells				

	Average		Red	
	-----		eyes	
	height	weight		

Males	1.9	0.003	40%	

Females	1.7	0.002	43%	

7 Références

- W3C <<http://www.w3.org/>>
- SGML <<http://www.sgml.org/>>
- XML <<http://www.w3.org/TR/xhtml1/>>
- *Un Nouveau Guide d'Internet (UNGI)* <http://www.imagnet.fr/ime/fr_ungi2.htm>
- *HTML4.01* <<http://www.w3.org/TR/1999/PR-html40-19990824/>>
- Apache <<http://www.apache.org/>>
- W3C <<http://www.w3.org/>>

8 Copyright

Copyright (c) 1999 *Emmanuel Fleury* <<mailto:fleury@lsv.ens-cachan.fr>>

Ce document peut être reproduit et distribué dans son intégralité ou partiellement, par quelque moyen physique que ce soit. Il reste malgré tout sujet aux conditions suivantes :

- La mention du copyright doit être conservée, et la présente section préservée dans son intégralité sur toute copie intégrale ou partielle.
- Si vous distribuez ce travail en partie, vous devez mentionner comment obtenir une version intégrale de ce document et être en mesure de la fournir.
- De portions de ce document peuvent être utilisées comme illustrations d'une présentation ou comme remarques sans autorisation préalable si les citations d'usage sont réalisées.