

# Développement Web : XHTML

But du document : Création de sites Web statique. Apprentissage du langage XHTML

1.	XHTML et CSS : les langages du Web .....	2
2.	Des navigateurs pour voir les pages web.....	2
3.	... et des éditeurs pour les créer !.....	3
4.	La solution de facilité : les éditeurs WYSIWYG.....	3
5.	La solution des Webmasters : les éditeurs de texte.....	3
6.	En résumé.....	3
7.	Généralités .....	3
8.	Codage des caractères(seulement en HTML pas en XHTML).....	4
9.	Les balises.....	5
10.	Structure d'un document HTML.....	6
	•Version de HTML.....	6
	•Les différents Doctypes .....	7
11.	Entête.....	7
12.	Titre .....	7
13.	Informations et directives.....	7
	•Meta http-equiv .....	8
	•Adresse de base.....	8
14.	Corps du document .....	9
15.	Codage des couleurs .....	9
16.	Paragraphe.....	10
17.	Titres.....	10
18.	Styles et attributs.....	10
19.	Saut de ligne.....	11
20.	Balises propriétaires.....	12
21.	Listes.....	12
22.	Les tableaux .....	13
23.	Les liens hypertextes .....	16
	•Les adresses URL .....	16
	•Les ancrs .....	17
24.	Les images .....	18
	•Les images mapées.....	18
25.	Les frames .....	19
	•Liens entre frames .....	21
	•Frames flottants.....	21
26.	Les formulaires.....	21
	•hamps <input>.....	21
	•Champ <SELECT>.....	22
	•Champ <TEXTAREA>.....	22
	•Champ <BUTTON>.....	22

Si vous voulez vous prétendre « webmaster » (c'est-à-dire créateur de site web), un minimum de culture générale est nécessaire.

Nous allons dans ce chapitre répondre aux questions suivantes :

- Que signifient les mots XHTML et CSS ?
- Quels sont les programmes qui permettent de naviguer sur un site ?
- Quels sont les programmes qui permettent d'en créer un ?

## 1. XHTML et CSS : les langages du Web

Pour créer un site web, on n'utilise pas un, mais deux langages !

- **XHTML** (*eXtensible HyperText Markup Language*) : autrefois connu sous le nom de HTML, c'est le langage de base du Web. En théorie, il est possible de faire tout un site rien qu'avec le langage XHTML. Toutefois, celui-ci ne serait pas très esthétique...
- **CSS** (*Cascading Style Sheets*) : c'est là qu'intervient le langage CSS. Son rôle est en quelque sorte de « décorer » votre site web, lui donner de l'allure. On utilise le CSS en particulier pour réaliser la mise en page du site, pour définir la police, la taille du texte, la couleur du texte et du fond, etc.

Au début, cela peut paraître un peu perturbant de se dire que l'on va avoir à manipuler deux langages. Pourtant, bientôt cela vous semblera naturel. En fait, ces deux langages permettent tout simplement de séparer le fond de la forme :

### **Du HTML au XHTML**

« Rome ne s'est pas faite en un jour »... Le Web non plus ! Celui-ci s'est développé au fur et à mesure, en fonction des idées et des possibilités offertes par les nouvelles technologies.

Ainsi le langage HTML a-t-il connu 4 versions apportant tour à tour leur lot d'améliorations. La dernière version en date est HTML 4.01. Depuis début 2000, le langage a changé de nom et s'appelle désormais XHTML car il est basé sur le langage XML. Il y a peu de différences entre HTML 4.01 et XHTML 1.0, si ce n'est qu'une plus grande rigueur est demandée pour des raisons d'uniformisation et pour respecter les règles du XML.

Voici une démonstration en images de ce que nous venons de voir : page avec XHTML seul et autre avec XHTML et CSS

Dans la première partie de ce cours (chapitres 2 à 5), nous ne travaillerons qu'avec le XHTML pour des raisons de simplicité (il serait dommage que tout se mélange dans votre tête dès le début !). Le site sera peu esthétique les premiers temps, mais la donne changera rapidement dès que nous introduirons le CSS à partir du chapitre 6.

## 2. Des navigateurs pour voir les pages web...

En l'espace de quelques années, le navigateur est devenu probablement le programme le plus utilisé sur un ordinateur. Et pour cause : c'est lui qui vous permet de surfer sur le Web !

Le rôle du navigateur est d'analyser le code XHTML et CSS des pages web et d'en produire un résultat visuel, facile à lire pour un humain (il faut avouer qu'un code source n'est pas très amusant à lire). Le problème est que les navigateurs interprètent parfois le XHTML et le CSS différemment, d'où l'intérêt de tester son site sur plusieurs navigateurs.

Vous savez probablement déjà ce qu'est un navigateur, mais pouvez-vous citer tous les navigateurs qui existent ? Personnellement en tout cas, je n'en suis pas capable car il en existe un très grand nombre, et je n'aurais pas la prétention de vous faire une liste complète. En revanche, je peux vous citer les plus connus d'entre eux :

- Internet Explorer (Windows) ;
- Mozilla Firefox (Windows, Mac et Linux) ;
- Opera (Windows, Mac et Linux) ;
- Safari (Mac) ;
- Konqueror (Linux), etc.

Il est impossible de tous les présenter ici, mais voyons quand même un peu plus en détail les trois premiers qui représentent à eux trois 97 % des internautes.

### 3. ... et des éditeurs pour les créer !

Nous savons que les navigateurs permettent de voir les sites web. Maintenant, nous nous rapprochons du sujet principal de ce livre : nous voulons connaître les programmes qui permettent d'en créer. Il existe un grand nombre d'éditeurs de sites web. Il faut savoir qu'on peut les regrouper en deux catégories :

- les éditeurs WYSIWYG ;
- les éditeurs de texte.

Nous allons voir ce qui différencie ces deux types d'éditeur.

### 4. La solution de facilité : les éditeurs WYSIWYG

WYSIWYG signifie : *What You See Is What You Get*, c'est-à-dire « ce que vous voyez est ce que vous obtenez ».

Ce type d'éditeur est le plus simple, du moins en apparence. Il vous permet de rédiger le contenu de votre site directement sans avoir à taper la moindre ligne de XHTML ou de CSS. Ces éditeurs fonctionnent un peu comme un traitement de texte tel que Word (qui est d'ailleurs lui même un éditeur WYSIWYG, mais de bien moindre qualité).

Le tableau 1-1 présente quelques éditeurs WYSIWYG parmi les plus connus.

### 5. La solution des Webmasters : les éditeurs de texte

Les éditeurs WYSIWYG sont des logiciels assez complexes et bien souvent très chers. En revanche, les éditeurs de texte sont pour la plupart gratuits et vous permettent de tout faire sur votre site web pour peu que vous connaissiez les langages XHTML et CSS (ça tombe bien, ce livre est là pour ça !). Voici une liste des éditeurs de texte que vous pouvez considérer,

### 6. En résumé...

Pour créer un site web, on doit utiliser deux langages que l'ordinateur comprend :

- **XHTML** : il permet de créer le contenu de votre site (c'est le *fond*).
- **CSS** : il permet de mettre en page votre site, de le décorer (c'est la *forme*).

Un navigateur est un logiciel qui permet de voir les sites web. Il analyse le code XHTML et CSS des pages et produit un résultat visuel. Ce résultat peut différer selon le navigateur que vous utilisez. Il est donc recommandé de tester continuellement son site sur au moins deux navigateurs à la fois pendant sa création, afin d'être sûr que tous vos visiteurs aient un résultat correct.

Pour créer un site, on doit utiliser un éditeur. Il en existe deux types :

- **Éditeurs WYSIWYG** : ils permettent de créer un site à la manière d'un traitement de texte. Ils génèrent automatiquement le code XHTML & CSS et sont donc plus faciles à prendre en main, mais limités.
- **Éditeurs de texte** : ce sont les plus puissants, bien que moins conviviaux au premier abord. Nous utiliserons un éditeur de texte tel que Notepad++ dans la suite de cet ouvrage pour réaliser notre site web.

### 7. Généralités

Le HTML (*HyperText Markup Language*) est le langage utilisé pour créer les documents (pages Web) reconnus par les navigateurs Internet (*browser*).

Contenu dans un simple fichier ASCII, il permet d'effectuer les mises en formes classiques (alignement des paragraphes, polices de caractères, choix

des couleurs, ...) offertes par les puissants traitements de textes du marché.

Le HTML peut également accueillir des images, des portions de codes issus d'autres langages voire des composants binaires afin d'enrichir le contenu des pages présentées.

Le HTML autorise d'autre part la définition de liens dits *hypertextes* entre des documents locaux ou distants ; c'est cette fonctionnalité qui permet de "naviguer" d'un document à un autre, par simple clic sur le lien choisi.

Indépendant de toute plate-forme, le document HTML sera lu sur n'importe quelle machine pourvue d'un navigateur adéquat.

## 8. Codage des caractères(seulement en HTML pas en XHTML)

Un fichier **HTML** est un fichier texte ASCII 7 bits, c'est à dire sans caractères accentués.

Pour exprimer un caractère accentué, il faut entrer une combinaison composée du caractère &, du code du caractère proprement dit et d'un ; (point virgule). Le code du caractère peut être exprimé soit par son code ASCII, précédé du signe #, soit par un mot clé prédéfini du HTML.

Ainsi é s'écrit `&eacute;` ou `&#233;`.

Voici une liste des principaux caractères spéciaux rencontrés dans la langue française :

Caractère	Code HTML	Code ASCII
à	<code>&amp;agrave;</code>	<code>&amp;#224;</code>
é	<code>&amp;eacute;</code>	<code>&amp;#233;</code>
è	<code>&amp;egrave;</code>	<code>&amp;#232;</code>
ê	<code>&amp;ecirc;</code>	<code>&amp;#234;</code>
ù	<code>&amp;ugrave;</code>	<code>&amp;#249;</code>

De même, les caractères spéciaux (courants en informatique) doivent être exprimés par des codes appropriés :

Caractère	Code HTML
>	<code>&amp;gt;</code>
<	<code>&amp;lt;</code>
&	<code>&amp;amp;</code>
"	<code>&amp;quot;</code>

Les retours chariots (passage à la ligne) sont ignorés par les navigateurs ; les sauts de ligne, sauf indication contraire, se font en fonction de la largeur de la fenêtre d'affichage.

De même, une suite de plusieurs espaces (blancs) et les tabulations ne seront pas prises en compte ; un espace insécable peut être inséré à l'aide du codage &nbsp;.

## 9. Les balises

Afin de distinguer les indications de mise en forme et le texte à afficher, le XHTML utilise des *balises* (aussi appelées *marqueurs* ou *tags*) indiquées entre les signes < et >.

Voici par exemple, le code d'une page XHTML très simple :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
<head>
  <title> Ma page html </title>
</head>
<body>
  Bienvenue dans le monde merveilleux du html !
</body>
</html>
```

<html>, <head>, <title> et <body> sont des balises ouvrantes tandis que </html>, </head>, </title> et </body> sont des balises fermantes ; la plupart du temps en effet, les balises fonctionnent par paires afin de préciser la portée de leurs actions. Une balise fermante est toujours composée par le nom de la balise ouvrante précédé du caractère /.

Il est parfois possible d'ajouter des attributs afin de préciser l'action d'une balise ; la balise prend alors la forme :

```
<Balise attribut_1 = valeur attribut_2 = valeur>
```

Par exemple :

```
<body bgcolor="#00FF00">
```

indique que le fond du document est d'une certaine couleur (*bgcolor* = *background color*)

### Remarque :

Un attribut s'écrit obligatoirement en **lettres minuscules** et ne comporte pas d'espace, tout comme le nom de la balise. Il est immédiatement suivi du signe égal « = » puis de guillemets qui entourent la valeur de l'attribut. La valeur de l'attribut n'est en revanche pas soumise aux mêmes règles : elle peut contenir des majuscules et des espaces sans problème.

En XHTML, les balises s'écrivent **entièrement en minuscules**, sans espace ni accent. En clair, on doit écrire <balise> et non <BALISE>. Écrire des balises en majuscules était autorisé auparavant en langage HTML. Le XHTML impose l'utilisation de minuscules pour des raisons d'uniformisation.

La rédaction d'une page XHTML peut se faire à l'aide d'un simple éditeur de texte ou bien en utilisant un outil adapté qui se chargera d'insérer les balises à votre place.

Dans la pratique, on utilise souvent les deux méthodes afin d'affiner le code de la page et d'insérer des balises non disponibles via l'outil.

## 10. Structure d'un document HTML

L'exemple précédent utilise les balises minimales pour la création d'un document HTML :

<code>&lt;html&gt; ... &lt;/html&gt;</code>	début/fin d'un document de type HTML
<code>&lt;head&gt; ... &lt;/head&gt;</code>	début/fin de la zone d'entête
<code>&lt;title&gt; ... &lt;/title&gt;</code>	début/fin du titre de la page
<code>&lt;body&gt; ... &lt;/body&gt;</code>	début/fin du document proprement dit

La balise (obligatoire) `<html>` indique simplement que le document est écrit en HTML. Cette balise doit figurer au tout début, tandis que le `</html>` terminant le document hypertexte se place en toute fin !

Entre les balises `<html>` et `</html>` se situent les deux parties composant l'ensemble du document HTML : l'entête et le corps.

L'entête est précisé grâce au couple `<head> </head>` et le corps par `<body> </body>`.

Les balises d'entête doivent toujours précéder les balises de corps.

L'insertion de commentaires est possible n'importe où grâce à la balise `<!--` qui ne nécessite pas de fermeture. Ainsi la ligne suivante ne sera pas interprétée par le navigateur :

```
<!-- Ceci est commentaire -->
```

Il est également possible (et plus courant) d'insérer un groupe de commentaires à l'aide des balises `<!-- et -->` :

```
<!--
```

```
Commentaires
```

```
-->
```

### Note :

Par la suite, les balises fermantes ne seront pas systématiquement indiquées.

- Version de HTML

La version du HTML de référence peut être indiquée (avant la partie en-tête `<HEAD>` avec la balise `<!DOCTYPE>`).

Ainsi la ligne

`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">` :impliquera l'utilisation du HTML en version 3.2

`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`:impliquera l'utilisation du XHTML en version 1

- **Les différents Doctypes**

Le Doctype indique la version de HTML ou de XHTML utilisée, mais aussi la rigueur du code. En effet, il y a plusieurs niveaux pour XHTML 1.0 :

- Strict ;
- Transitional ;
- Frameset.

Le niveau Strict est celui que l'on recommande d'utiliser dans la mesure du possible, les autres étant surtout utilisés pour faire la transition entre un « vieux » code en HTML et un code XHTML.

Nous utiliserons donc le niveau Strict qui implique le plus de rigueur, mais qui sera aussi à l'avenir le seul niveau autorisé (XHTML 1.1 est obligatoirement en Strict par exemple).

## 11. **Entête**

La zone d'entête `<head> ... </head>`, prologue au document proprement dit, contient des informations destinées principalement au navigateur et qui ne seront donc pas à proprement parlé, visible par l'utilisateur.

## 12. **Titre**

Le titre du document est indiqué grâce à la balise `<title>`.

`<title>` Bienvenue sur mon site Internet `</title>`

Si la zone d'entête ne contient qu'une balise, il s'agit en général de celle ci. Ce titre apparaît habituellement dans la barre de titre du navigateur.

## 13. **Informations et directives**

Il est possible de faire figurer au sein de l'entête un certain nombre d'informations sur le document qui seront lues et interprétées par le serveur HTTP.

Ces informations sont précisées à l'aide de la balise `<meta>` suivie d'attributs appropriés (associés à `name` ou `http_equiv`) et valorisées à l'aide du mot clé `content`.

On recense trois catégories d'informations :

- les informations destinées au navigateur (directives)
- les informations utiles pour le référencement
- les informations réservées à l'utilisateur
  
- META NAME

On peut, à l'aide de la balise `<meta name = "... " content = "... ">` indiquer des informations telles que le nom de l'auteur (*Author*), un "droit de copie" (*Copyright*), le public visé (*Rating*), ...

```
<meta name = "author" content ="thierry cosnuau">
```

```
<meta name = "copyright" content ="1999 thc ">
```

```
<meta name = "rating" content ="general">
```

L'attribut *Keywords* sera utilisé par les moteurs de recherche pour référencer votre page comme traitant des sujets indiqués par les mots clés successifs :

```
<meta name = "keywords" content ="mot_clé1,mot_clé2,mot_clé3">
```

Cette indication pourra éventuellement être complétée par l'attribut *Description* :

```
<meta name = "description" content ="description de la page">
```

- **Meta http-equiv**

La balise `<meta http-equiv = "... " content = "... ">` permet d'indiquer une directive au navigateur.

L'attribut *Expires* est utilisée par la plupart des navigateurs pour donner la date à partir de laquelle la page ne doit plus être conservée dans le cache du navigateur mais bien rechargée sur le serveur lors d'un nouveau passage sur la page. La valeur 0 indique que la page doit être systématiquement rechargée.

```
<meta http-equiv= "expires" content ="sun, 05 sep 1999 12:00:00 gmt">
```

Le chargement automatique d'une page précisée par son URL après un certain délai se fait à l'aide de l'attribut **Refresh**. C'est notamment utile lorsque les sites changent de place ...

```
<meta http-equiv= "refresh" content ="10;URL=PAGE2.HTM">
```

Le pays (et donc la langue) d'origine de la page est déclaré par l'attribut *Content-language*(FR pour la France).

L'adresse email du concepteur, utile pour lui envoyer des remarques concernant le site ou la page, est valorisée par l'attribut *Reply-to*.

Un grand nombre d'attributs peuvent être employés avec META NAME ou META HTTP-EQUIV. L'utilisateur curieux de les connaître se connectera au site des organismes de normalisation tels que le *W3 consortium* ([www.w3.org](http://www.w3.org)).

- **Adresse de base**

La balise `<base href =...>` donne la base de l'adresse URL (*Uniform Resource Locator*) qui sera placée devant les références relatives dans le document, de telle sorte que, hors contexte, les fichiers soient cherchés à l'adresse `Adresse_de_base/Adresse_relative`.

Ainsi :

```
<base href=http://www.tcosnuau.fr/>
```

permettra de référencer les adresses courtes telles que *mapage.htm* à l'adresse *http://www.tcosnuau.fr/mapage.htm*

#### 14. Corps du document

Le corps du document, commencé par la balise **<body>**, constitue la partie visible de la page XHTML, c'est à dire les éléments apparaissant dans le navigateur de l'utilisateur.

A l'intérieur de cette zone, le texte incorporé sans balises apparaîtra tel quel dans la page.

La balise **<body>** peut elle même comporter un certain nombre d'attributs tels que la couleur de fond ou l'image utilisée comme "papier peint" de la page.

```
<body bgcolor = "#AABBCC">
```

ou

```
<body background = "monfond.gif">
```

On peut également préciser la couleur des différents composants du document ; l'exemple suivant affichera le texte en bleu-gris, les liens **link** en vert, les liens visités **vlink** en jaune et les liens actifs **alink** en cyan :

```
<body text="#AABBCC" link="#00FF00" vlink="#FFFF00" alink="#00FFFF">
```

#### 15. Codage des couleurs

La couleur peut être précisée soit par un code hexadécimal, soit par un nom prédéfini reconnu par le HTML.

La méthode de codage hexadécimal est moins lisible mais plus précise. Elle permet en effet de préciser, sur une échelle allant de 0 à 255, les valeurs des composantes rouge, verte et bleue : #RRVVBB (RGB en anglais).

On peut utiliser les couleur de la manière suivante : `rgb(0,0,255)`

Exemple : `<body text= " rgb(0,0,255) " >`

- Codage des principales couleurs

Couleur	Code hexadécimal	Code HTML
rouge	#FF0000	red
vert	#00FF00	lime
bleu	#0000FF	blue
blanc	#FFFFFF	white
noir	#000000	black

jaune	#FFFF00	yellow
cyan	#00FFFF	cyan

Ce codage des couleurs, valable pour la couleur de fond, restera valable pour l'ensemble des attributs précisant la couleur.

## 16. Paragraphe

Le début et la fin d'un paragraphe de texte sont indiqués respectivement par les balises `<p>` et `</p>`.

Un paragraphe contient en général une portion de texte d'allure homogène.

L'attribut **align** permet de préciser l'alignement (**center**, **left** ou **right**).

`<p align="right"> Ce texte est très droit. </p>`

La fin d'un paragraphe est marquée par un retour à la ligne.

## 17. Titres

Six niveaux de titres (*Headers*) peuvent être utilisés au sein du document HTML. Les titres sont entourés par les tags `<hn>` et `</hn>` où *n* prend une valeur comprise entre 1 et 6, 1 correspondant au titre principal.

L'attribut align permet comme pour les paragraphes de préciser l'alignement.

`<h1 align="center"> Titre de mon document </h1>`

`<h2> Sous titre </h2>`

`<h6> Titre de petit niveau </h6>`

## 18. Styles et attributs

Les balises précisant le style et les attributs des caractères sont :

<code>&lt;b&gt;</code>	gras [ <i>Bold</i> ]
<code>&lt;strong&gt;</code>	gras (logique)
<code>&lt;i&gt;</code>	italique [ <i>Italic</i> ]
<code>&lt;em&gt;</code>	italique (logique) [ <i>Emphasis</i> ]
<code>&lt;u&gt;</code>	soulignés [ <i>Underline</i> ]
<code>&lt;s&gt;</code> ou <code>&lt;strike&gt;</code>	barrés
<code>&lt;pre&gt;</code>	texte formaté (avec espaces et sauts de ligne)
<code>&lt;big&gt;</code>	grande taille
<code>&lt;small&gt;</code>	petite taille
<code>&lt;font size=<i>nn</i>&gt;</code>	taille absolue (1-7)

<code>&lt;font size=+n&gt;</code>	taille relative (+ ou -)
<code>&lt;font color = "#nnnnnn"&gt;</code>	couleur
<code>&lt;font face=police&gt;</code>	Type de police (Arial, Courier, Serif, ...)
<code>&lt;sub&gt;</code>	mise en indice
<code>&lt;sup&gt;</code>	mise en exposant
<code>&lt;center&gt;</code>	centré

D'autre part, un certain nombre de styles prédéfinis peuvent être employés pour clarifier la présentation d'une page :

<code>&lt;code&gt;</code>	Code de programmation
<code>&lt;samp&gt;</code>	Exemple [ <i>Sample</i> ]
<code>&lt;var&gt;</code>	Variable
<code>&lt;def&gt;</code>	Définition
<code>&lt;address&gt;</code>	Adresse
<code>&lt;cite&gt;</code>	Citation (courte)
<code>&lt;blockquote&gt;</code>	Citation (longue)
<code>&lt;tt&gt;</code>	Police de taille fixe [ <i>Type Text</i> ]
<code>&lt;kbd&gt;</code>	Machine à écrire [ <i>Keyboard</i> ]

### Note :

La balise `<blockquote> ... </blockquote>` peut être utilisée pour effectuer un ou plusieurs retraits.

## 19. Saut de ligne

Un retour à la ligne s'incorpore à l'aide de la balise `<br>` (*Break*).

Une ligne horizontale peut être insérée avec la balise `<hr>` (*Horizontal Rule*) dont les attributs précisent éventuellement la taille et la position.

<code>&lt;hr size=n&gt;</code>	épaisseur
<code>&lt;hr width=n/%&gt;</code>	longueur (absolue / pourcentage)
<code>&lt;hr align=left/right/center&gt;</code>	position
<code>&lt;hr noshade&gt;</code>	sans ombre

Par défaut, la ligne occupe toute la largeur et son épaisseur vaut 2.

Les deux balises `<br>` et `<hr>` ne nécessitent pas de balises fermantes.

Il est également possible d'empêcher le retour à la ligne réalisé par défaut lorsqu'une ligne dépasse la largeur de l'écran à l'aide des

balises `<nobr> ... </nobr>`. La césure (éventuelle) peut alors être indiquée à l'aide de la balise `<wbr>` (Word break).

## 20. Balises propriétaires

Certaines balises de mise en forme ne fonctionnent que pour un certain type de navigateur ; leur emploi est donc fortement déconseillé, puisqu'il prive une partie des clients de l'effet souhaité.

Citons pour mémoire la balise propriétaire de Microsoft `<marquee>` permettant de faire défiler du texte sous Internet Explorer mais qui ne fonctionnera que sous ce navigateur et la balise propriétaire de Netscape `<blink>` qui fera clignoter le texte, uniquement sous Navigator ...

(La balise `<marquee>` admet les attributs `behavior=slide / scroll`, `direction` ou encore `scrollldelay=n`)

## 21. Listes

Les listes à puces (**Unordered List**) sont incorporées au sein des tags `<ul>` et `</ul>` ; chaque item de la liste (**List Item**) est ajouté à l'aide de la balise `<li>` (sans contrepartie fermante).

```
<ul>
  <li> Pomme
  <li> Poire
  <li> Citron
</ul>
```

La représentation des puces peut être modifiée à l'aide de l'attribut `STYLE` qui peut prendre les valeurs `Disc`, `Square` ou `Circle` (prise en compte uniquement par certains navigateurs).

Les listes numérotées (**Ordered List**) sont placées entre les balises `<ol>` et `</ol>`, chaque item étant ajouté avec la balise `<li>` :

```
<ol>
  <li> Un
  <li> Deux
  <li> Trois
</ol>
```

Il est possible de préciser la manière d'exprimer la numérotation à l'aide de l'attribut `type` dans la balise `<ol>` ou dans la balise `<li>` :

`type = 1` Numérotation classique en chiffres arabes (par défaut)

`type = I` Chiffres romains en majuscules

`type = i` Chiffres romains en minuscules

**type** = a Lettres minuscules

**type** = A Lettres majuscules

On peut également faire démarrer la numérotation à partir d'un certain rang avec l'attribut **start** de la balise OL.

Les listes de définition (*Definition List*) peuvent être ajoutées entre les balises **<dl>** et **</dl>** avec, pour chaque définition le terme à définir (*Definition Term*) placé entre **<dt>** et **</dt>** et la définition proprement dite (*Definition Data*) entre **<dd>** et **</dd>**.

```
<dl>
  <dt> Pomme </dt>
    <dd> Fruit A Compote </dd>
  <dt> Orange </dt>
    <dd> Fruit A Jus </dd>
</dl>
```

**Note :**

Des listes peuvent également être mise en place avec les balises **<menu>** et **<dir>** qui fonctionnent sensiblement de la même façon que **<ul>** et qui sont donc peu utilisées.

**22. Les tableaux**

En HTML, les tableaux sont utilisés non seulement pour afficher des données en colonnes mais également pour aligner les éléments les uns par rapport aux autres, en masquant la bordure du tableau.

Un tableau est composé de lignes et de colonnes qui forment les cellules du tableau. Toutefois, le nombre de colonnes du tableau sera déduit du nombre de cellules ajoutées pour chaque ligne.

La balise HTML indiquant l'insertion d'un tableau est **<table>**.

Une ligne (*Table Row*) est débutée par **<tr>** et terminée par **</tr>**.

Une cellule (*Table Data*) est délimitée par **<td>** et **</td>**.

On utilise éventuellement **<th>...</th>** pour insérer une cellule d'entête.

Le titre d'un tableau peut être précisé avec la balise **<caption>...</caption>** qui doit se situer entre les balises **<table>** et **</table>** ; ce titre sera placé, selon les versions du navigateur au dessus ou au dessous du tableau.

- Exemple

```

<table>

  <tr><th>Ville</th><th>Pays</th><th>Continent</th></tr>
  <tr><td>Paris</td><td>France</td><td>Europe</td></tr>
  <tr><td>Turin</td><td>Italie</td><td>Europe</td></tr>
  <tr><td>Caire</td><td>Egypte</td><td>Afrique</td></tr>

</table>

```

La balise `<table>` possède les attributs suivants :

<code>&lt;table border = n&gt;</code>	taille de la bordure du cadre
<code>&lt;table cellspacing = n&gt;</code>	espace entre les cellules
<code>&lt;table cellpadding = n&gt;</code>	espace entre le bord et le contenu
<code>&lt;table width = n/%&gt;</code>	largeur du tableau (absolue/pourcentage)
<code>&lt;table bgcolor = "#nnnnnn"&gt;</code>	couleur de fond
<code>&lt;table bordercolor = "#nnnnnn"&gt;</code>	couleur de bordure
<code>&lt;table bordercolordark = "#nnnnnn"&gt;</code>	couleur de bordure sombre
<code>&lt;table bordercolorlight = "#nnnnnn"&gt;</code>	couleur de bordure claire
<code>&lt;table align =left/right/center &gt;</code>	alignement

Note :

Les attributs BORDERCOLORLIGHT et BORDERCOLORDARK sont prioritaires par rapport à BORDERCOLOR.

Les attributs des balises de ligne `<TR>` et de cellule `<TD>` permettent également d'affiner l'apparence :

<code>&lt;tr align = ...&gt;</code>	alignement de la ligne
<code>&lt;td align = ... &gt;</code>	alignement de la cellule
<code>&lt;td colspan = n&gt;</code>	étirement sur n colonnes
<code>&lt;td rowspan = n&gt;</code>	étirement sur n lignes
<code>&lt;td width = n/%&gt;</code>	largeur de la cellule (absolue/pourcentage)
<code>&lt;td height = n&gt;</code>	hauteur de la cellule
<code>&lt;td bgcolor = "#nnnnnn"&gt;</code>	couleur de fond
<code>&lt;td bordercolor = "#nnnnnn"&gt;</code>	couleur de bordure
<code>&lt;td bordercolordark = "#nnnnnn"&gt;</code>	couleur de bordure sombre
<code>&lt;td bordercolorlight = "#nnnnnn"&gt;</code>	couleur de bordure claire

Note :

Les attributs de <TD> peuvent également être appliqués à <TH>.

- Exemple complet

```
<table border=3 width=100%  
  
    bordercolor=green  
    bordercolorlight=red  
    bordercolordark=blue  
    bgcolor=yellow  
    cellspacing=4  
    cellpadding=6  
  
>  
  
    <caption>g&eacute;ographie</caption>  
  
    <tr align=center>  
        <th width=33%>ville</th>  
        <th width=33%>pays</th>  
        <th width=33%>continent</th>  
    </tr>  
  
        <tr align=center>  
            <td>paris</td>  
            <td>france</td>  
            <td>europe</td>  
        </tr>  
  
        <tr align=center>  
            <td>turin</td>  
            <td>italie</td>  
            <td>europe</td>  
        </tr>  
  
        <tr align=center>  
            <td>caire</td>  
            <td>egypte</td>  
            <td>afrique</td>  
        </tr>  
  
        <tr align=center>  
            <td colspan=2>luxembourg</td>  
            <td>europe</td>  
        </tr>  
  
</table>
```

### 23. Les liens hypertextes

Par essence le HTML est un langage hypertexte qui permet à l'utilisateur de naviguer entre les pages par simple clic souris sur un mot, généralement souligné et d'une couleur différente, appelé lien.

Les liens permettent d'aller :

- vers un autre endroit au sein du document courant (signet)
- vers une autre page du site visité (lien local)
- vers un autre site (lien distant)

Il est également possible de créer un lien vers une adresse email ou encore un serveur FTP.

- Les adresses URL

Une adresse URL (*Uniform Resource Locator*) indique la localisation de l'information demandée par le navigateur.

Une URL est composée de plusieurs parties :

*protocole*  
*nom de l'hôte*  
*répertoire*  
*fichier*

Ainsi pour l'adresse

*http://www.microsoft.com/ie/welcome.htm*  
*http* représente le protocole  
*www.microsoft.com* représente le nom de l'hôte  
*ie* représente le répertoire  
*welcome.htm* représente le fichier (page)

Les différents protocoles sont :

HTTP ( <i>HyperText Transfer Protocol</i> )	lien distant vers une page web
FTP ( <i>File Transfert Protocol</i> )	lien distant vers un fichier
MAILTO	lien vers un mail
FILE	Lien vers un fichier

Une URL ne peut être composée que des caractères alphanumériques classiques (A-Z, a-z, 0-9) et les symboles dollar (\$), tiret (-), souligné (\_) et point (..).

Les autres caractères doivent être précisés à l'aide de codages spéciaux indiqués à l'aide du caractère pour cent (%) suivi du caractère ASCII du caractère en notation hexadécimale.

Ainsi %20 représente l'espace, %3f le point d'interrogation ...

- **Les ancrs**

La balise exprimant un lien ou ancre (*Anchor*) est **<A>**.

Pour définir un signet, utile pour se déplacer à l'intérieur du document, on utilise la balise **<A NAME=...>**

Pour atteindre un emplacement qu'il soit local ou distant on utilise la balise **<A HREF = ...>**.

De la valorisation de l'attribut HREF, dépendra le type de déplacement.

- Exemple de signets

```
<h1>sommaire</h1>
<a href="#part1">première partie</a>
<br>
<a href="#part2">seconde partie</a>
<br>
<hr width="75%">
<br>
<p>
<a name="part1">ici commence la partie 1</a>
<!-- texte -->
</p>
<p>
<a name="part2">ici commence la partie 2</a>
<!-- texte -->
</p>
```

- Exemple de lien local

```
<A HREF="PAGE2.HTM"> cliquez ici pour la page 2 </A>
```

- Exemple de lien local avec signet

```
<A HREF="PAGE2.HTM#TEST"> cliquez ici pour la page 2 </A>
```

- Exemple de lien distant

```
<A HREF="HTTP://WWW.MICROSOFT.COM"> Site Microsoft </A>
```

- Exemple de lien vers un mail

```
<A HREF="MAILTO:tcosnuau@free.fr"> Envoyez moi un mail </A>
```

- Exemple de lien vers un fichier

```
<A HREF="DOWNLOAD/SAMPLE.ZIP"> Fichier exemple </A>
```

## 24. Les images

En plus du texte, une page est composée la plupart du temps d'images qui la rende plus attractive et aussi ... plus lente à afficher ; il convient donc d'utiliser ces "décorations" à bon escient afin de ne pas trop alourdir le site et donc décourager l'éventuel visiteur.

Les images ne font pas partie intégrante du document ; seules leurs références sont indiquées. Les fichiers images doivent donc être disponibles sur le serveur HTTP, aux emplacements indiqués.

Le Web utilise deux formats graphiques satisfaisants du point de vue rapport "qualité d'image / taille du fichier". Ils comportent chacun 256 couleurs

Ces deux formats sont le **JPEG** (*Join Point Expert Group*) et le **GIF** (*Graphic Interchange Format*).

Les images GIF ont la particularité de pouvoir être animées, c'est à dire que l'image affichée évoluera au cours du temps afin de recréer une petite animation.

L'insertion d'une image au sein d'une page HTML se fait à l'aide du tag **<IMG>** utilisé conjointement avec un certain nombre d'attributs. Ainsi pour préciser la source, c'est à dire le nom du fichier image, on emploie l'attribut SRC.

- Exemple simple

```
<IMG SRC="IMAGE.GIF">
```

On peut éventuellement définir un texte alternatif qui s'affichera si le navigateur ne reconnaît pas les images, avec l'attribut ALT.

La dimension de l'image peut être précisée avec les attributs WIDTH et HEIGHT.

Une bordure peut être ajoutée avec BORDER=taille.

L'alignement par rapport au texte se fait avec l'attribut ALIGN dont les valeurs possibles sont *Top*, *Middle*, *Bottom*, *Left* et *Right*. D'autres valeurs permettent d'affiner cet alignement : *TextTop*, *AbsMiddle*, *AbsBottom* et *BaseLine*.

L'espace entre l'image et un objet peut être précisée par les valeurs de VSPACE (vertical) et HSPACE (horizontal).

- Exemple complet

```

```

- Les images mapées

Une image peut être découpée en zones afin de permettre l'association de liens multiple en fonction de la zone cliquée. On parle alors de *Maps* et de "liens hypergraphiques".

L'utilisation d'un logiciel graphique dédié est fortement conseillée pour définir les différentes zones de l'image (*MapThis*, *MapEdit*, ...).

Il faut préciser l'attribut `USEMAP="#nom"` lors de l'insertion de l'image avec la balise `<img>` et ajouter une balise `<map name = "nom">` définissant les zones de l'image et leurs références.

Ces zones sont définies grâce à la balise `<area>` contenue dans une structure `<map> ... </map>`.

La forme de la zone peut être rectangulaire (**rect**), circulaire (**circle**) ou polygonale (**polygon**).

L'attribut `COORDS` précise les coordonnées qui permettront au navigateur de reconstituer la forme géométrique. Par convention, le point d'origine (0,0) est situé au coin supérieur gauche de l'image ; l'axe des abscisses (X) est alors dirigé vers la droite et l'axe des ordonnées (Y) vers le bas.

- Exemple

```

```

```
<map name="map123">
```

```
<area shape=rect coords="1,1,55,50" href="page1.htm" alt="un">
```

```
<area shape=rect coords="57,1,110,50" href="page2.htm" alt="deux">
```

```
<area shape=rect coords="1,60,110,120" href="page3.htm" alt="trois">
```

```
<area shape=default href="page0.htm">
```

```
</map>
```

## 25. Les frames

Le langage HTML permet de diviser l'écran en plusieurs fenêtres alors appelées "cadres" (*frames*), dans le but d'afficher plusieurs documents simultanément. Cela est notamment utile pour présenter une table des matières qui restera tout au long de la visite du site.

Le fichier qui décrit les cadres est un document HTML assez simple dont la partie `<body> ... </body>` est remplacée par des balises spécifiques :

```
<frameset> ... </frameset>.
```

L'agencement des fenêtres se fait à l'aide des attributs **rows** (horizontal) et **cols** (vertical) de la balise `<frameset>`.

On précise à l'aide des ces attributs la taille de chacune des zones.

Il est possible d'utiliser le caractère \* pour compléter une suite de zone à 100% (par exemple `rows="25%,40%,*"` équivaut à `rows="25%,40%,35%"`)

La balise `<frame src= ...>` permet de préciser le document HTML "classique" qui sera placé à l'intérieur du cadre.

- Exemple

```
<frameset rows = "25%,75%">
```

```
<frame src="page1.htm">
  <frameset cols = "40%,60%">
    <frame src="page2.htm">
    <frame src="page3.htm">
  </frameset>
</frameset>
```

La bordure d'une FRAME ou d'un FRAMESET peut être supprimée en valorisant l'attribut FRAMEBORDER à "No" ; la taille de la bordure peut être indiquée dans ces deux balises avec l'attribut BORDER ; la couleur de la bordure sera précisée par l'attribut BORDERCOLOR.

Il est également possible :

- d'indiquer l'espacement avec l'attribut FRAMESPACING.
- de préciser si une barre de défilements (*ScrollBar*) doit apparaître avec l'attribut SCROLLING dont les valeurs possibles sont "YES", "NO" ou "AUTO".
- d'empêcher le re-taillage des bordures par l'attribut NORESIZE.

Enfin, il est utile de nommer les *frames* pour faciliter la navigation entre les différents écrans. Cette propriété est précisée par l'attribut NAME.

- Exemple

```
<frameset rows="80,*" frameborder="yes" framespacing="1">
  <frame src="azurtitl.htm" name="titre" scrolling="no" noresize
  marginheight="0" marginwidth="0" framespacing="0"
  frameborder="No">
  <frameset cols="180,*">
    <frame src="azursomm.htm" name="sommaire" scrolling="no"
    frameborder="0" framespacing="0">
    <frame src="azurfirs.htm" name="main" frameborder="0"
    framespacing="0">
  </frameset>
</frameset>
```

Si le navigateur ne supporte pas les *frames*, il convient de prévoir une section **<NOFRAMES>** dans laquelle sera affiché un message d'avertissement

- Exemple

```
<NOFRAMES>
```

Votre navigateur ne supporte pas les frames

```
</NOFRAMES>
```

- **Liens entre frames**

Les liens de la balise <A> doivent contenir l'attribut TARGET précisant dans quel cadre se situera la page visée.

Cet attribut sera valorisé avec le nom du cadre (NAME) ou avec une valeur prédéfinie :

_blank	indique au navigateur qu'il doit créer une nouvelle fenêtre
_self	indique que la nouvelle page sera chargée à la place de la page courante
_top	entraîne l'affichage de la page sur l'ensemble de la fenêtre

- **Frames flottants**

Disponible à partir de la version 4 de HTML, les frames flottants offrent la possibilité d'inclure à l'intérieur même d'une page une zone pointant vers une autre page.

Cette insertion est réalisée à l'aide de la balise <IFRAME> dont l'attribut SRC, tout comme pour <FRAME> précise la page à inclure.

Les attributs de <IFRAME> sont sensiblement les mêmes que ceux employés pour <FRAME>.

## 26. **Les formulaires**

La première forme d'interactivité des pages HTML est fournie par l'utilisation des formulaires qui reprennent certains des contrôles habituellement utilisés par les interfaces graphiques des systèmes d'exploitation actuels.

L'utilisateur peut ainsi remplir un *champ de saisie*, préciser ses choix à l'aide de *cases à cocher* et de *radio boutons* et transmettre ses données à l'aide de *boutons d'action*.

L'insertion d'un formulaire dans un document HTML se fait avec la balise <form>.

Le nom du formulaire est indiqué par l'attribut name.

Le type d'action (get ou post) est précisé avec l'attribut method.

L'action est indiquée grâce à l'attribut action.

Trois types de contrôles, correspondant à trois balises peuvent être inclus dans un formulaire.

- **Champs <input>**

Les contrôles "classiques" sont insérés à l'aide du tag `<input type = ...>` dont les valeurs possibles sont :

<b>text</b>	zone de saisie
<b>password</b>	zone de saisie remplacée par des *
<b>hidden</b>	zone cachée
<b>checkbox</b>	case à cocher
<b>radio</b>	radio bouton
<b>submit</b>	bouton poussoir d'envoi
<b>reset</b>	bouton poussoir de réinitialisation
<b>image src="..."</b>	bouton image

Les attributs qui peuvent être précisés sont :

<b>name</b>	nom du contrôle
<b>value</b>	valeur par défaut d'un champ de saisie
<b>checked</b>	indique si un contrôle RADIO ou CHECKBOX est actif
<b>size</b>	taille du champ pour les champs caractères (par défaut 20)
<b>maxlength</b>	nombre maximum de caractères
<b>submit</b>	bouton poussoir d'envoi
<b>reset</b>	bouton poussoir de réinitialisation
<b>image src="..."</b>	bouton image

- **Champ <SELECT>**

Une liste déroulante peut être insérée à l'aide de la balise `<SELECT>`.

Les différents lignes sont précisées par des balises `<OPTION>`.

- **Champ <TEXTAREA>**

Une zone de texte multi-lignes peut être introduite par la balise `<TEXTAREA>`.

- **Champ <BUTTON>**

Un bouton quelconque peut être ajouté avec la balise `<BUTTON>`.

- Exemple 1

```
<p>voici une forme :</p>
<p>&nbsp;</p>
<form method="post" name="testform">
<p>zone de texte à saisir : <input type="text" size="20">
```

```
name="t1"></p>
<p>cases à cocher :</p>
<p><input type="checkbox" name="c1">option 1</p>
<p><input type="checkbox" name="c2">option 2</p>
<p>choix multiples</p>
<p><input type="radio" name="r1" value="v1">choix 1</p>
<p><input type="radio" checked name="r1" value="v2">choix 2</p>
<p>bouton de commande : <input type="submit" name="b1"
value="envoyer"></p>
</form>
```

- Exemple 2

```
<form>
votre syst&eagrave;me : <br>
<select name="system">
<option>mac os
<option selected>pc windows
<option>unix
</select> <p>
logiciels utilisés : <br>
<select name="software" multiple size=5>
<option> frontpage
<option selected> internet explorer
<option selected> xara 3d
<option> xara web style
</select> <p>
<textarea name="commentaires" rows=20 cols=60>
n&eacute;ant
</textarea><p>
<button onclick="javascript:alert('hello')">
sayhello
</button>
</form>
```

La mise en page des formulaires se fait usuellement à l'aide de tableau (<TABLE>) dont on ne fait pas apparaître les bordures.

Les formulaires sont surtout utilisés conjointement avec des éléments tels que le langage JavaScript ou les programmes CGI.

Ils permettent alors d'établir un véritable dialogue entre l'utilisateur et le serveur contenant le site.

- Exemple d'interaction avec du JavaScript

```
<html>
```

```
<head>
<title>test form & javascript</title>
<script language="javascript">
function hello()
{
document.write("bonjour <b>" +
document.forms[0].elements["ef_nom"].value + "</b>");
}
</script>
</head>
<body>
<form method="get" action="javascript:hello();">
entrez votre nom : <input type="text" name="ef_nom">
<input type="submit" value="bonjour">
</form>
</body>
</html>
```