

# **Cours HTML**

## Sommaire

1	Introduction.....	3
2	Préambule.....	3
3	Environnement.....	4
4	Choix et Historique.....	5
5	Internet.....	6
5.1	Environnement Client-Serveur.....	6
5.2	Moteur de recherche.....	6
5.3	Le navigateur.....	6
6	Du concret !.....	7
6.1	La première page HTML.....	7
6.2	Mise en forme.....	8
6.2.1	Les couleurs.....	8
6.2.2	Police et taille.....	8
6.2.3	Centré, gras.....	9
6.2.4	Paragraphes, titres.....	9
6.2.5	Listes.....	10
6.2.6	Séparateur.....	10
6.3	Chemins d'accès aux fichiers.....	10
6.3.1	Insérer une image.....	11
6.3.2	Liens hypertexte.....	11
6.3.3	Insérer une vidéo ou du son.....	12
6.4	Tableaux.....	13
6.4.1	Création.....	13
6.4.2	Mise en forme.....	14
6.5	Les formulaires.....	15
6.5.1	Le plus simple.....	15
6.5.2	Listes, cases à cocher.....	16
6.6	Les cellules (Frames).....	17
6.6.1	Un exemple néanmoins.....	18
7	Les feuilles de style.....	19
8	Editeurs HTML.....	20
Annexe 1	: Sources d'informations.....	21
Annexe 2	: Comparaison HTML-XML.....	22
Annexe 3	: Les scripts.....	23
Annexe 4	: Résumé des balises.....	24

# 1 Introduction

Ce cours a pour objectif de présenter les éléments de base du langage HTML.

Après une présentation de l'environnement du web et de l'historique du web, l'essentiel du cours portera sur la création de pages HTML présentant ces différents éléments de base.

## 2 Préambule

- Quels sont les différents moyens pour afficher un texte à l'écran ?
- Comment est présenté un texte ?
- Comment indique-t-on le style des caractères, leur format, leur couleur ... ?
- Quelle est la taille du fichier ?

### 3 Environnement

Html émane du **World Wide Web Consortium**, abrégé **W3C**. C'est un consortium fondé en octobre 1994 pour promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, CSS, PNG, SVG et SOAP. Le W3C n'émet pas des normes au sens européen, mais des recommandations à valeur de standards industriels.

Quelques abréviations :

- SGML : Standard Generalized Markup Language (langage normalisé de balisage généralisé) – est un langage de description à balises, de norme ISO (ISO 8879:1986).
- HTML : l'Hypertext Markup Language est le langage informatique créé et utilisé pour écrire les pages Web. HTML permet en particulier d'insérer des hyperliens dans du texte, donc de créer de l'hypertexte, d'où le nom du langage.
- XML : Extensible Markup Language, « langage de balisage extensible ») est un langage informatique de balisage générique. Le W3C recommande XML pour exprimer des langages de balisages spécifiques (exemples : XHTML, SVG, XSLT). Son objectif initial est de faciliter l'échange automatisé de contenus entre systèmes d'informations hétérogènes, notamment, sur Internet. XML est un sous-ensemble de SGML dont il retient plusieurs principes dont : la structure d'un document XML est définissable (DTD) et validable par un schéma, un document XML est entièrement transformable dans un autre document XML. Cette syntaxe est reconnaissable par son usage des chevrons (< >), elle s'applique à de plus en plus de contenus.
- XHTML est le successeur de HTML (de l'anglais HyperText Markup Language), XHTML respectant la syntaxe définie par XML, plus récente et plus simple que la syntaxe définie par SGML respectée par HTML.
- CSS : Cascading Style Sheets, ce sont les feuilles de style en cascade utilisées pour décrire la présentation d'un document structuré écrit en HTML ou en XML
- DTD : La Document Type Definition, ou Définition de Type de Document, est un document permettant de décrire un modèle de document SGML ou XML
- PNG : Portable Network Graphics est un format d'images numériques *ouvert*, qui a été créé pour remplacer le format *propriétaire* GIF, dont la compression était soumise à un brevet.
- SVG : Scalable Vector Graphics est un format de fichier basé sur XML permettant de décrire des ensembles de graphiques vectoriels.
- SOAP : Simple Object Access Protocol est un protocole de RPC orienté objet bâti sur XML.
- RPC : Remote Procedure Call est un protocole permettant de faire des appels de procédures sur un ordinateur distant à l'aide d'un serveur d'application (modèle client-serveur)

## 4 Choix et Historique

- **L'hypertexte** : l'idée est de réaliser automatiquement le renvoi qu'un texte un peu élaboré fait à d'autres parties de lui-même ou à d'autres textes. Cela se fait en activant des zones particulières du document, les liens hypertexte (hyperlinks). On fait remonter cette idée à un article de 1945, qui a frappé des imaginations qui l'ont mise en œuvre au SRI (Stanford research institute) et au Xerox Parc. On l'a trouvée un peu plus tard chez Apple avec Hypercard (mais au sein d'une seule application sur une machine, pas à travers des réseaux interconnectés). Une bonne idée a sans doute été de décliner une catégorie de documents (très) simple à partir de SGML (Standard generalized markup language), un système de balisage généralisé permettant de construire de façon très ouverte des structures dans du texte. Des séquences particulières de caractères codent, au sein même du texte, sa structure. L'HTML est une catégorie de texte SGML, spécifiée, standardisée par une DTD (Document type definition). La simplification permettait d'avoir des interpréteurs viables, qui puissent assurer un fonctionnement interactif.
- **L'interopérabilité** : permettre aux documents HTML d'être accessibles sur des plates-formes et des navigateurs différents est fondamental dans la conception de ce langage. L'ambition était d'avoir un langage de description de documents au format ouvert qui soit indépendant de ce qui servirait à lire les pages. Par conséquent, il n'est pas prévu que telle ou telle police de caractère soit utilisée explicitement, ni que le moniteur du client utilise telle ou telle résolution. Le logiciel exploité pour la navigation devrait lui aussi ne pas être une source de discrimination.  
Parvenir à l'interopérabilité permet de baisser les coûts des fournisseurs de contenus parce qu'ils n'ont à développer qu'une seule version d'un document. Si cet effort n'est pas fait, le risque est grand de voir le Web régresser vers un monde propriétaire de formats incompatibles, en réduisant en fin de compte son potentiel commercial au détriment de tous ses participants.  
Le langage HTML a été développé avec l'intuition que les appareils de toutes sortes devaient pouvoir utiliser les informations sur le Web : les ordinateurs personnels avec des écrans de résolution et de profondeur de couleurs variables, les téléphones cellulaires, les appareils portables, les appareils de synthèse et de reconnaissance de la parole, les ordinateurs avec une bande passante faible comme élevée, et ainsi de suite.
- **Séparer la structure de la présentation** : L'expérience montre que la séparation de la structure d'un document de sa présentation permet de réduire les coûts de diffusion vers un grand éventail de plates-formes, de médias, etc., et facilite la mise à jour des documents.

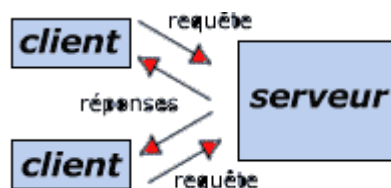
Différentes versions de HTML ont été implémentées. La version la plus à jour actuellement est HTML 4.

## 5 Internet

Appelé aussi le web, la toile ...

### 5.1 Environnement Client-Serveur

Les applications sur le web fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante, qui leur fournit des services, l'accès aux sites ...



### 5.2 Moteur de recherche

Il existe une énorme quantité d'informations sur Internet (plusieurs millions ou milliards de documents, sur les milliers ou millions de serveurs), et ces informations sont pour la plupart renouvelées quotidiennement. Le moteur de recherche est un élément indispensable pour s'y retrouver !

Un moteur de recherche (Searchbot en anglais) est une machine spécifique (matérielle et logicielle) chargée d'indexer des pages web afin de permettre une recherche à l'aide de mots-clés dans un formulaire de recherche.

Les principaux : AltaVista, Google, Yahoo ...

### 5.3 Le navigateur

L'élément clé pour la navigation au sein des pages web est le navigateur (en anglais browser). Utilisé sur le client, ce logiciel est capable d'interroger les serveurs web, d'exploiter leurs résultats et de mettre en page les informations grâce aux instructions contenues dans la page HTML.

Les deux principaux navigateurs utilisés sur Internet :

- Mozilla Firefox,
- Microsoft Internet Explorer.

ATTENTION : malgré la volonté du W3C d'uniformiser les langages pour assurer une compatibilité optimale entre tous les intervenants sur le web. Les implémentations des normes ne sont jamais parfaites. Lors de l'affichage d'une page html, il est donc fréquent de ne pas obtenir exactement le même comportement suivant le navigateur utilisé.

## 6 Du concret !

HTML est un langage à balises. Les documents écrits dans un langage à balises contiennent essentiellement deux choses : des données, et des meta-données. Pour faire simple : la meta-donnée est une information supplémentaire qui ajoute un contexte ou une signification particulière à une donnée. Un exemple, prenons le texte :

```
Ma première page en HTML
```

```
Bonjour, voici ma première PAGE web !
```

Un humain sait que « Ma première page en HTML » est le titre du texte, et « Bonjour, voici ma première page web ! » son contenu avec des polices de caractères différentes. Toutefois, les programmes d'ordinateurs ne sont pas humains et ne savent pas tout ça. Nous utilisons donc des meta-données exprimées par des balises pour ajouter un sens à la donnée.

Un fichier HTML doit être suivi de l'extension « .html » ou « .htm ».

### 6.1 La première page HTML

Exemple, créons le fichier page1.html avec ce contenu :

```
<HTML>
<HEAD>
<TITLE>Ma première page en HTML</TITLE>
</HEAD>

<BODY>
Bonjour, voici ma première page web !

</BODY>
</HTML>
```

Ouvrons le fichier avec un navigateur et regardons ce qu'il affiche.

Explications des balises utilisées :

- `<html>` et `</html>` : la première informe votre navigateur qu'il va afficher une page HTML. Cette balise doit être la première de votre code. La seconde est la balise de fin. Elle referme la première. (remarquez le / devant le mot contenu entre crochets). Elle doit être la dernière de votre code HTML et signifie la fin de votre code HTML.
- `<head>` et `</head>` : la première ouvre l'entête de votre page. Cet entête comprend pour l'instant le titre de votre page. Cette balise est toujours placée juste après la balise `<html>`. La seconde, referme l'entête de votre page (remarquez encore une fois le / devant le mot entre crochets).
- `<title>` et `</title>` : la première ouvre la partie titre de votre page. Ce titre sera placé dans la barre tout au-dessus de votre navigateur. La seconde referme la première encore une fois en plaçant un / devant le mot entre crochets. La balise "title" doit être ouverte puis refermée à l'intérieur de la balise "head". C'est à dire qu'il faut que la balise "title" doit être ouverte puis refermée avant que la balise "head" ne soit refermée.
- `<body>` et `</body>` : la première ouvre le début du corps de votre document. C'est là que vous allez placer textes, images et tout ce que contiendra votre page. Cette balise doit se situer après la balise de fermeture de l'entête (=> `</head>`). La seconde referme la

première et maintenant je suppose que vous avez compris le principe pour fermer une balise...

## 6.2 Mise en forme

### 6.2.1 Les couleurs

Ajoutez dans page1.html entre les balises « BODY » :

```
<FONT color="red">Rouge</FONT> <BR>
<FONT color="#FFFF00">Jaune</FONT> <BR>
<BODY bgcolor="#000000">
<FONT color="#FFFFFF">texte blanc sur fond noir</font>
```

Et actualisons la page dans le navigateur.

Explications des nouvelles balises utilisées :

- `<br>` : cette balise permet de faire un saut de ligne. Lorsque vous écrivez du texte dans une page web, il y a un retour automatique à la ligne lorsque le texte atteint le bord droit de l'écran. Avec la balise `<br>`, vous pourrez faire des sauts de ligne à votre guise. Un "BR" correspond à un saut de ligne, pour laisser une ligne blanche entre 2 paragraphes par exemple il faut placer 2 "BR" (==> `<BR><BR>`). Il n'y a pas de balise de fin !
- `<font>` et `</font>` : elles permettent de modifier l'apparence du texte. Elle doit recevoir des informations provenant d'attribut(s) pour connaître la mise en forme à appliquer. Pour l'instant, nous avons juste vu l'attribut "color" qui permet de mettre le texte en couleur. Les couleurs les plus fréquentes peuvent être nommées explicitement en anglais (red, green, blue,...) sinon il faut placer le code Hexadécimal de la couleur voulue. Essayez "green", "#008000", "blue", "#0000FF", "yellow", "#FFFF00". Vous trouverez la liste des principaux codes couleurs sur notre page Codes couleurs. La couleur du texte est par défaut noir (#000000 ou black).
- Nous avons également vu l'attribut "bgcolor" dans `<BODY>` : "bgcolor" (==>background color : couleur de fond) permet de mettre le fond de l'écran en couleur tout comme le fait "color" pour le texte. C'est le même principe pour le code des couleurs. La couleur du fond est par défaut blanc (#FFFFFF).

Petites "définitions" :

- Un attribut est un paramètre qui fournit des indications à une balise. Il possède un nom et très souvent une valeur à préciser (nom = "valeur"). Il se rajoute dans la balise de début, c'est à dire toujours entre les `< >` après le nom de la balise (par exemple : `<font color="#ff0000">`).
- Le code Hexadécimal qui permet de définir une couleur est composé de 6 caractères précédés de #. Ces 6 caractères définissent la "teneur" en rouge, vert et bleu (les couleurs de bases) de la couleur à former. Il n'y a que 16 caractères possibles : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

### 6.2.2 Police et taille

Commentez ou retirez `<BODY bgcolor="#000000">` puis ajoutez :

```
<FONT size=7>Taille 7</FONT> <BR>
<FONT size="-2">Taille -2</FONT> <BR>

<FONT size=4 face="Verdana">Taille 4 en Verdana</FONT> <BR>
```



```
<FONT size=2 face="Comic sans MS">Taille 3 en Comic sans MS</FONT> <BR>
<FONT face="Arial, Times New Roman" color="#336699">Taille normal en
Arial si la police existe sinon en Times New Roman en couleur
#336699</FONT> <BR>
```

Explications des nouveaux attributs utilisés :

- l'attribut "size" : "size" (==> taille) permet de modifier la taille du texte compris entre <font> et </font>. Vous avez le choix entre donner un nombre entre 1 et 7 (par défaut => 3) ou bien donner une valeur de "décalage" par rapport à la taille par défaut du navigateur ("-2", "-1", "+1", "+2", "+3", "+4").
- l'attribut "face" : "face" (==> apparence) permet de choisir la police dans laquelle le texte sera affiché. Vous pouvez préciser plusieurs noms de police séparés par des ','. Si le navigateur ne reconnaît pas le premier nom, il essaiera avec le suivant... S'il ne reconnaît aucun nom, il utilisera la police par défaut du navigateur utilisé.

Petites remarques :

- Vous pouvez tout à fait cumuler les attributs dans une balise (séparés par un espace) !
- Inutile d'utiliser des polices très spéciales car si les visiteurs ne l'ont pas sur leur ordi, le navigateur ne la reconnaîtra pas et donc ne l'utilisera pas.

### 6.2.3 Centré, gras ...

Ajoutons :

```
<B>texte en ???</B> <BR>
<CENTER>texte centre</CENTER> <BR>
```

Explications des nouvelles balises utilisées :

- <b> et </b> : écrit le texte en gras.
- <i> et </i> : écrit du texte en italique. en gras.
- <u> et </u> : souligne du texte. Evitez d'utiliser trop cette balise car un texte souligné peut être confondu avec un lien et ça peut être gênant pour le visiteur.
- <center> et </center> : centre le texte par rapport à l'écran.
- <!-- et --> : Ces balises permettent de commenter votre code. Le commentaire peut s'étendre sur plusieurs lignes.

Remarques importantes :

- Attention à bien refermer vos balises sinon la suite du texte aura également les propriétés de la balise non refermée. Par exemple si vous oubliez de refermer la balise pour placer des commentaires (<!-- ... -->), la fin de votre code ne s'affichera pas !
- On peut cumuler les balises, pour par exemple mettre un texte en gras, en italique et centré. Il faut bien faire attention à l'ordre des balises pour les refermer (dans l'ordre inverse de l'ouverture) !

### 6.2.4 Paragraphes, titres

Ajoutons :

```
<P align="right">texte aligné</P> <BR>
<BLOCKQUOTE>texte en retrait</BLOCKQUOTE> <BR>
<H1>Titre en 1</H1>
```

Explications des nouvelles balises utilisées :

- `<p>` et `</p>` avec l'attribut "align" : permet de créer un paragraphe avec du texte aligné soit à gauche ("left"), à droite ("right"), au centre ("center") ou en justifié ("justify").
- `<blockquote>` et `</blockquote>` : permet de décaler un bloc de texte par rapport au bord gauche.
- `<h1>` et `</h1>` : permet de définir des titres. Il existe 6 types de titres numérotés de 1 à 6. Vous pouvez utiliser le même attribut "align" que celui de `<p>`.

## 6.2.5 Listes

Ajoutons :

```
<UL type="disc">
<LI>Page 1</LI>
<LI>Page 2</LI>
</UL>
<br>
<OL>
<LI>Page 3</LI>
<LI>Page 4</LI>
<LI>Page 5</LI>
</OL>
```

Explications des nouvelles balises utilisées :

- `<ul>` et `</ul>` avec l'attribut "type" : indique le début et la fin d'une liste de puces (disc, circle, square).
- `<ol>` et `</ol>` : indique le début et la fin d'une liste numérotée (avec des lettres : attribut type="a").
- `<li>` : désigne un nouvel élément d'une liste.

## 6.2.6 Séparateur

Ajoutons :

```
<HR size=10 align="right" color="#FF0000" width="60%"> <BR>
```

Explications de la nouvelle balise utilisée :

- `<hr>` : permet de placer un trait séparateur. Plusieurs attributs possibles : "width" pour en préciser la largeur soit en pixels soit en pourcentage, "size" pour l'épaisseur, "color" pour lui donner de la couleur, "align" pour l'aligner soit à gauche ou à droite. Pas de balise de fin !

## 6.3 Chemins d'accès aux fichiers

Nous allons à dans les prochains chapitres avoir besoin d'autres fichiers, par exemple des fichiers images (\*.gif, \*.jpg), d'autres pages web (pour les liens hypertextes),...

Quel chemin emprunter pour accéder à ce fichier ?

Il y a deux types de chemins d'accès : les chemins absolus et les chemins relatifs :

- Les chemins relatifs permettent d'accéder à des fichiers qui se trouvent sur la même machine que le fichier (la page web) qui les appelle.

- Pour utiliser un fichier image qui se trouve dans le même répertoire que votre page web, il suffira simplement de préciser le nom de ce fichier :  
nom\_du\_fichier.ext (le .ext remplace l'extension du fichier dans cet exemple théorique, dans la pratique, nous aurons .gif, .html, .js,...)
- Si le fichier se trouve dans un sous-répertoire du répertoire qui contient notre page web, il faudra accéder au fichier comme ceci : nom\_du\_sous\_rep/fichier.ext
- Si le fichier se trouve dans un autre répertoire, voici comment procéder :  
../nom\_autre\_rep/fichier.ext  
Le "../" permet de remonter d'un "étage" dans l'arborescence des répertoires.  
Si vous devez remonter de 2 "étages" => ../../nom\_du\_rep/fichier.ext
- Les chemins absolus utilisent l'URL du fichier, donc pour accéder à des fichiers qui se situent sur une autre machine (pour les liens hypertextes vers des sites extérieurs par exemple), il faut utiliser des chemins absolus et donc des URLs (par exemple :  
[http://www.un-site.com/un\\_rep/sous\\_rep/fichier.ext](http://www.un-site.com/un_rep/sous_rep/fichier.ext))

Définition : Une URL (Uniform Resource Locator) est un format de nommage universel pour désigner une ressource sur Internet.

### 6.3.1 Insérer une image

Ajoutons dans notre fichier :

```
<CENTER><IMG src="image1.gif" border=10 width=88 height=31
alt="Logo"></CENTER>
```

Explications de la nouvelle balise utilisée :

- `<img>` : insère une image dans votre page. Il faut préciser le chemin (ou adresse) de l'image à ajouter au moyen de l'attribut "src" (obligatoire !). Il n'y a pas de balise de fin !  
Les attributs "width" et "height" permettent respectivement de spécifier la largeur et la hauteur de l'image en nombre de pixels.  
L'attribut "border" désigne la taille du bord à afficher autour de l'image (par défaut : 0).  
Et enfin "alt" (comme alternative) permet de préciser un texte qui sera affiché si l'image n'est pas trouvée (=> croix rouge) ou bien si vous laissez votre souris dessus.

Quelques remarques :

- Ce dernier attribut "alt" est très important pour le référencement de votre site. Il vaut donc mieux ne pas le négliger et l'utiliser dans chacune de vos balises `<IMG>`.
- Si vos images ne s'affichent pas (une croix rouge les remplace), c'est certainement une erreur dans le chemin d'accès vers le fichier image. Vérifier bien le nom du fichier, le nom du répertoire ... Il arrive également que les certains éditeurs HTML vous insèrent des chemins du style `c:/mes documents/...`, les images s'afficheront donc en local (chez vous) mais par sur le web. C'est normal vu que ce chemin n'existe pas sur la machine où se trouvent vos pages web. Il faudra donc modifier ces chemins par des chemins relatifs.
- Pour mettre une image comme fond d'écran, il faut utiliser l'attribut "background" de la balise `<BODY>`. Il vaut toujours mieux utiliser un fond d'écran fixe (pas une image animée).

### 6.3.2 Liens hypertexte

Les liens portent bien leur nom. Ils permettent de faire la liaison entre les pages web. Pour changer de page, il faut cliquer sur un lien, le navigateur chargera alors la nouvelle page.

Il y a 3 types de liens :

- Les liens internes à un site (=> utilisation très fréquente des chemins relatifs car plus court à taper).
- Les liens internes à une page (ou ancre) (=> utilisation de chemins absolus ou relatifs).
- Les liens externes (=> uniquement des chemins absolus !)

Ajoutons d'abord quelque part au milieu de notre page, la définition d'une ancre :

```
<BR> <CENTER><FONT color="green" size=4><A name="ancre1">Renvoi  
ici</A></FONT></CENTER> <BR> <BR>
```

Puis à la fin, avant </body> :

```
<A href="page2.html">Lien vers la page 2</A> <BR>  
<A href="#ancre1">Lien vers l'ancre 1</A> <BR>  
  
<A href="http://www-cs-faculty.stanford.edu">Lien vers un site  
externe</A> <BR>  
<A href="http://www-cs-faculty.stanford.edu/~knuth/taocp.html#vol4  
>Lien vers une ancre sur une page d'un site externe</A> <BR>  
<A href="page2.html">  
</A> <BR>
```

Explications des nouvelles balises utilisées :

- <a> et </a> : permettent de créer un lien hypertexte. Son attribut principal est "href". En effet, il permet de préciser l'adresse de la page de destination du lien qui sera créé. Cette adresse peut être soit absolue soit relative.  
Si vous souhaitez faire un lien interne à une page, il faut créer une ancre.

Quelques remarques :

- Si le contenu d'une page est assez long, un lien interne à la page avec une ancre pourrait être très pratique pour revenir dans le haut de la page (sans utiliser les barres de scroll). Au début de votre texte, vous créez un ancre sur le premier mot du texte et à la fin du texte, vous rajoutez un lien vers cette ancre.
- Dans nos exemples, nous avons uniquement utilisé du texte entre les balises <a> et </a>. Mais rien ne vous empêche d'utiliser une image plutôt que du texte !  
<a href="page3.html"></a>
- Il existe également un attribut "target" qui vous permet par exemple d'ouvrir un lien dans une nouvelle fenêtre comme ceci : <a href="URL" target="\_blank">lien</a>  
Par défaut, un lien s'ouvre dans la même page que la page où se trouve le lien, hors si vous avez des liens vers des sites extérieurs, si un visiteur clique sur un de ces liens, le site va s'ouvrir dans la fenêtre où se trouve votre site (il ne sera donc plus visible puisque remplacé par la nouvelle page). Il peut donc être utile d'utiliser l'attribut target pour les liens vers des sites externes pour que votre site reste visible également.  
Pour des liens internes à votre site, il est souvent déconseillé d'ouvrir 36 fenêtres différentes avec les pages de votre site pour ne pas troubler les visiteurs.
- Il faut également que les cibles des liens existent, on peut créer des fichiers contenant juste du texte pour page2.html et page3.html.

### 6.3.3 Insérer une vidéo ou du son

Si vous voulez insérer une vidéo ou du son dans votre page, il y a deux possibilités; soit en ouvrant le lecteur, soit en l'intégrant directement dans la page.

Pour l'ouvrir hors de la page avec windows media player il suffit de faire un lien vers la vidéo de la même façon que l'on insert un lien.

Pour l'insérer directement dans la page, exemple :

```
<embed src="mavideo.mpg" border="1" width="300" height="300" loop="1"
controls="false" autostart="true" autoload="true">
```

Ou pour du son :

```
<embed src="mamusique.wma" volume="75">
```

Explications de la nouvelle balise utilisée :

- `<embed>` avec l'attribut `src` (obligatoire) spécifiant l'url de la vidéo. Ses attributs sont nombreux :
  - `border` : De 1 à 6 spécifie la bordure.
  - `width` et `height` : Largeur et hauteur de "l'écran", en pixels.
  - `loop` : Spécifie le nombre de fois qu'est jouée la vidéo (pour infini, mettre -1).
  - `controls` : Active ou désactive la présence des boutons (lecture, pause, stop...) sous l'image "true" (défaut) pour oui et "false" pour non.
  - `autostart` : Active (true : défaut) ou désactive (false) le démarrage automatique. Si les boutons ne sont pas activés ne mettez pas false à cet attribut au quel cas votre visiteur ne pourrait pas voir la vidéo.
  - `autoload` : Active (true : défaut) ou désactive (false) le chargement automatique de la vidéo.

Remarque :

- Attention lors des développements dans le répertoire « Documents and Settings » : ce nom de répertoire contient des espaces qui peuvent être mal interprétés et faire échouer les liens. Placez vous de préférence dans un répertoire dont le nom ne contient pas d'espaces.

Pour insérer un fond sonore à sa page, testons :

```
<bgsound src="http://site.voila.fr/slimp4/son.mid" loop="1">
```

Explications de la nouvelle balise utilisée :

- `<bgsound>` possède seulement deux attributs:
  - `src` Indique l'url du morceau (obligatoire)
  - `loop` Spécifie le nombre de nombre de fois qu'est joué le morceau (-1 pour infini)

## 6.4 Tableaux

### 6.4.1 Création

Ajoutons pour notre premier tableau :

```
<BR><BR>
<TABLE border=1>
<TR>
<TD width=200>Cellule 1</TD>
<TD>Cellule 2</TD>
</TR>
<TR>
<TD>Cellule 3</TD>
<TD width=60%>Cellule 4</TD>
```

```
</TR>
</TABLE>
```

Explications des nouvelles balises utilisées :

- `<table>` et `</table>` : création d'un tableau. Nous avons dans notre exemple précisé, au moyen de l'attribut "border" (0 par défaut), la taille de la bordure autour des cellules de notre tableau. (la balise de fin est obligatoire !)
- `<tr>` : débute une nouvelle ligne dans la tableau. La balise de fin (`</TR>`) n'est pas obligatoire. Chaque fois que le navigateur tombera sur une balise `<TR>`, il considérera que la ligne précédente est terminée et commencera une nouvelle. Toutes les lignes d'un même tableau doivent contenir le même nombre de cellules.
- `<td>` : crée une nouvelle cellule dans un tableau. Cette cellule doit être contenue dans une ligne et donc une balise "TR" doit être déjà ouverte... Vous pouvez ensuite créer autant de cellules que vous voulez dans un même ligne. Une cellule peut contenir un peu tout et n'importe quoi. On peut y mettre du texte, des images ... La cellule va s'adapter automatiquement aussi bien en largeur qu'en hauteur au contenu le plus haut et le plus large. Et pour finir, toutes les cellules d'une même colonne ont la même largeur (celle de la cellule la plus large), idem pour la hauteur des cellules d'une même ligne. Comme pour "TR", la balise de fin (`</TD>`) est optionnelle.

Remarque :

- Même si les balises de fin `</td>` et `</tr>` sont optionnelles, il vaut quand même mieux les utiliser pour éviter des problèmes de compatibilité ou de mise en page avec certains vieux navigateurs.

## 6.4.2 Mise en forme

Ajoutons :

```
<BR><BR>
<TABLE border=2 bgcolor="#000000" cellpadding=5 cellspacing=3
align="center">
<TR>
<TD width=100 height=100 bgcolor="#CCCCCC" valign="top"
align="left">#CCCCCC</TD>
<TD width=100 height=100 bgcolor="#FFFFFF" valign="bottom"
align="center">#FFFFFF</TD>
<TD rowspan=2 bgcolor="#333333"><FONT color="#0099FF">Fusion de 2
lignes</FONT></TD>
</TR>
<TR>
<TD bgcolor="#FFFCC" colspan=2 valign="middle" align="center">Fusion de
2 colonnes</TD>
</TABLE>
```

Explications des nouveaux attributs utilisés :

- "bgcolor" permet de donner un fond de couleur à un tableau ou à une cellule.
- "width" permet de préciser la largeur d'un tableau ou d'une cellule. La largeur peut être donnée en pixel ou en pourcentage.
- "height" permet de donner la hauteur d'une cellule. Il n'est pas possible de préciser la hauteur d'un tableau. Attention, si vous donnez une certaine hauteur X à une cellule et que vous placez par exemple une image dont la hauteur est plus grande, la cellule s'adaptera à la taille de l'image en "oubliant" son attribut hauteur ! (idem en largeur).
- "cellpadding" définit l'espace en pixel entre le bord d'une cellule et son contenu. (1 par défaut).
- "cellspacing" définit l'espace entre les cellules du tableau. (1 par défaut).

- "colspan" permet de faire fusionner des cellules d'une même ligne. Il faut y préciser le nombre de cellules à fusionner.
- "rowspan" permet de faire fusionner des cellules d'une même colonne. Il faut y préciser le nombre de cellules.
- "align" précise l'alignement sur un axe horizontal du tableau ou du contenu de la cellule. (valeur possible : "left" par défaut, "right", "center").
- "valign" précise l'alignement du contenu des cellules sur un axe vertical. (valeurs possible : "middle", "bottom", "top").

Petits plus :

- Vous pouvez mettre une image comme fond d'un tableau à la place d'une simple couleur (comme pour la balise BODY) avec l'attribut background.
- Il est également possible de changer la couleur de la bordure autour du tableau avec l'attribut "bordercolor" (par exemple bordercolor="#000000").

## 6.5 Les formulaires

Les formulaires permettent de récupérer des informations auprès de vos visiteurs. Vous pouvez ainsi créer une certaine interactivité avec vos visiteurs. Ils pourront vous envoyer des messages et vous pourrez mieux les connaître.

Il y a plusieurs moyens pour récupérer ces informations. Soit vous utilisez un script coté serveur (voir Annexe 3), ce script est interprété sur un serveur chez votre hébergeur et pourra traiter les informations. Soit vous pouvez recevoir les infos directement sur votre adresse email au moyen d'un "mailto".

### 6.5.1 Le plus simple

Ajoutons encore à notre fichier :

```
<BR><BR>
<FORM action="mailto:vous@votre-domaine.com" method="post"
enctype="text/plain">
  <INPUT type="hidden" name="intro" value="Voici les infos récoltées via
la formulaire : ">
  Entrez votre nom : <INPUT type="text" name="nom" size=25>
  <BR>
  Entrez l'adresse de votre site : <INPUT type="text" name="url" size=40
value="http://">
  <BR>
  Entrez un mot de passe : <INPUT type="password" name="pass" size=10
maxlength=10>
  <BR>
  <INPUT type="submit" value="Envoyer" name="submit">   <INPUT
type="reset" value="Effacer les données" name="reset">
</FORM>
```

Explications des nouvelles balises utilisées :

- <form> et </form> : elles délimitent un formulaire. La présence de la balise de fin est primordiale !  
L'attribut "enctype" permet de spécifier l'encodage des données.  
L'attribut "action" désigne l'adresse du script (CGI, PHP) qui va traiter les données. Ici on utilise l'envoi vers un mail : action="mailto:adresse\_email@domaine.com".  
L'attribut "method" spécifie la manière dont les données remplies dans le formulaire

doivent être transmises. Il peut prendre 2 valeurs : "POST" ou "GET". Avec GET, les données sont envoyées au serveur via l'URL et sont donc visibles dans celle-ci ! Avec POST, les données sont directement envoyées au serveur et ne sont pas visibles. On utilisera presque toujours la méthode POST pour les formulaires.

- `<input>` : cette balise permet de placer un champ dans lequel les informations peuvent être recueillies. Pas de balise de fin !

L'attribut "name" permet de désigner un champ du formulaire par un nom pour le traitement des données. Chaque nom doit être unique !

L'attribut "type" spécifie le type de champ. Il peut prendre plusieurs valeurs :

- "text" pour entrer du simple texte
- "hidden" pour un champ caché non visible par le visiteur (on ne peut pas y ajouter du texte !)
- "password" pour entrer un mot de passe (les caractères sont remplacés par des \*)
- "submit" pour créer un bouton pour envoyer les données du formulaire.
- "reset" pour créer un bouton qui vide le formulaire et le remet comme au chargement de la page.

L'attribut "value" permet de donner une valeur à un champ avant que le visiteur ne commence à le remplir. Dans le cas d'un bouton (submit ou reset), cette valeur sera le texte affiché sur le bouton !

L'attribut "size" donne la largeur du champ (20 par défaut).

L'attribut "maxlength" quant à lui définit le nombre maximal de caractères pouvant être tapés dans un champ. Si vous essayez de taper un caractère supplémentaire, il ne sera pas pris dans le champ.

## 6.5.2 Listes, cases à cocher

Ajoutons maintenant :

```
<BR><BR>
```

```
<FORM>
```

```
Comment trouvez-vous ce cours jusqu'à présent :
```

```
<SELECT name="liste" size="3"></option>
```

```
<OPTION value="excellent">Excellent</option>
```

```
<OPTION value="bon" selected>Bon</option>
```

```
<OPTION value="moyen">Moyen</option>
```

```
<OPTION value="faible">Faible</option>
```

```
<OPTION value="nul">Pitoyable</option>
```

```
</SELECT>
```

```
</FORM>
```

```
<BR><BR>
```

```
<FORM>
```

```
Entrez votre texte : <BR>
```

```
<TEXTAREA name="texte" cols=40 rows=5>Texte de départ</TEXTAREA>
```

```
</FORM>
```

```
<BR><BR>
```

```
<FORM>
```

```
Quels sont vos hobbies : <BR>
```

```
Le sport : <INPUT type="checkbox" name="hobby1" value="sport" checked>
```

```
<BR>
```

```
Internet : <INPUT type="checkbox" name="hobby2" value="internet"> <BR>
```

```
La télévision : <INPUT type="checkbox" name="hobby3" value="tv" checked>
```

```
</FORM>
```

```
<BR><BR>
```

```
<FORM>
```



```

Quel est votre niveau en HTML : <BR>
<INPUT type="radio" name="niveau" value=3>Excellent <BR>
<INPUT type="radio" name="niveau" value=2 checked>Moyen <BR>
<INPUT type="radio" name="niveau" value=1>Débutant
</FORM>

```

Explications des nouvelles balises utilisées :

- <select> et </select> : crée une liste déroulante. On peut lui attribuer un nom avec l'attribut "name" ainsi que le nombre d'éléments visibles de la liste avec "size" (1 par défaut).
- <option> et </option> : ajoute un élément à la liste déroulante créée avec un SELECT. On peut lui donner une valeur grâce à son attribut "value". Par défaut, c'est la première balise option qui sera celle par défaut. Si vous voulez que ce soit une autre, il faut ajouter le mot suivant dans la balise en question : 'selected'.
- <textarea> et </textarea> : elles délimitent une zone de texte. Comme toujours on peut la nommer avec "name". On peut également en spécifier la taille avec "rows" (nombre de lignes) et "cols" (nombre de colonnes).
- Nous avons également vu 3 nouvelles valeurs pour l'attribut type d'un INPUT :
  - "button" : crée un bouton cliquable. On lui donne un nom au moyen de "name" et une valeur (qui sera le texte affiché sur ce bouton) avec "value". Les simples boutons n'ont pas un grand intérêt à première vue mais ils en auront beaucoup quand on leur affectera des actions via un JavaScript.
  - "checkbox" : crée une case à cocher. On peut bien entendu lui donner un nom et une valeur mais on peut aussi décider si elle doit être cochée ou non au chargement de la page. Si la case doit être cochée, il faut rajouter dans la balise le mot suivant : 'checked'. Par défaut une case n'est pas cochée.
  - "radio" : crée un bouton radio. Un bouton radio seul n'a pas un grand intérêt. En général, on en place plusieurs pour donner un choix aux visiteurs. Contrairement aux cases à cocher avec lesquelles on peut sélectionner plusieurs valeurs, avec les boutons radio, on ne peut en choisir qu'une ! Quand on choisit une autre valeur, celle qui était sélectionnée ne l'est plus.

ATTENTION : les boutons radios doivent avoir le même nom (via l'attribut "name") !

Pour sélectionner la valeur par défaut, comme pour les cases à cocher, il faut ajouter le mot 'checked' dans la balise du bouton par défaut.

## 6.6 Les cellules (Frames)

Très pratiques au début d'Html, l'utilisation des frames (voir aussi <iframe>) est aujourd'hui controversée. Dans la logique de départ, les frames servaient à structurer l'interface d'une page Internet de manière à avoir plusieurs éléments distincts pouvant défiler indifféremment les uns des autres. Bien vite cette balise fût détournée pour afficher d'autres informations non voulues par le visiteur, allant même jusqu'à masquer son contenu (taille à 0 par exemple).

De plus les frames ne sont pas compatibles avec tous les navigateurs. Prenons comme exemple, les navigateurs en mode texte. On dira "mais personne ne les utilise !" ... personne sauf les déficients visuels ou autres handicapés. D'autres catégories de navigateurs existent, citons par exemple PocketIE pour Windows CE, la taille de l'écran n'étant pas toujours adapté à la lecture d'une page comportant des frames, c'est pourquoi ces navigateurs ne les reconnaissent pas !

Pour ceux qui iront plus loin, la solution consiste en fait à utiliser les balises DIV.

### 6.6.1 Un exemple néanmoins

Les frames ne sont pas compatibles avec <BODY>, il faut créer un fichier à part :

```
<HTML>
<FRAMESET COLS="25%,*">
<FRAME SCROLLING="yes" NAME="cel-g" SRC="partie1.html">
  <FRAMESET ROWS="50%,50%">
    <FRAME SCROLLING="yes" NAME="cel-d-h" SRC="partie2.html">
    <FRAME SCROLLING="yes" NAME="cel-d-b" SRC="partie3.html">
  </FRAMESET>
</FRAMESET>
<NOFRAME>
Dommage, votre <I>Browser</I> ne permet pas les <I>frames</I>.
</NOFRAME>
</HTML>
```

## 7 Les feuilles de style

### Définition :

Charte graphique : son but est de conserver une cohérence graphique dans les réalisations graphiques d'une même organisation, projet ou entreprise quels que soient les différents intervenants de la production (graphiste, directeur artistique ...).

L'intérêt de réaliser une charte graphique est double :

- L'identité graphique reste intacte quelles que soient les réalisations graphiques, afin que l'organisation, le projet ou l'entreprise (qui sont une somme d'individualités) parle «d'une seule voix». Les anglophones parlent de corporate design (littéralement : dessin de l'entreprise).
- La charte graphique permet au récepteur (par opposition à l'émetteur) d'identifier facilement cet émetteur et, par habitude si la cohérence est respectée, de se repérer visuellement dans les différentes réalisations graphiques - y compris au sein même d'un même support graphique - facilitant sa lecture.

Le concept de feuilles de style est apparu en 1996 avec la publication par le W3C d'une nouvelle recommandation intitulée « Cascading StyleSheets » (feuilles de style en cascade), notée CSS.

Le principe des feuilles de style consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments. Il suffit de définir par un nom un ensemble de définitions et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte. Il est ainsi possible de créer un groupe de titres en police Arial, de couleur verte et en italique.

Les feuilles de style ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation. En effet, le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent. Grâce aux feuilles de style, lorsque la charte graphique d'un site composé de plusieurs centaines de pages web doit être changée, il suffit de modifier la définition des feuilles de style en un seul endroit pour changer l'apparence du site tout entier !

Elles sont appelées « feuilles de style en cascade » (en anglais « Cascading Style Sheets ») car il est possible d'en définir plusieurs et que les styles peuvent être hérités en cascade.

Les feuilles de style permettent notamment :

- d'obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à une même définition de style ;
- de permettre le changement de l'aspect d'un site complet entier par la seule modification de quelques lignes ;
- une plus grande lisibilité du HTML, car les styles sont définis à part ;
- des chargements de page plus rapides, pour les mêmes raisons que précédemment ;
- un positionnement plus rigoureux des éléments.

Vous trouverez plusieurs cours sur les feuilles de style sur internet :

<http://www.netalya.com/fr/Css-intro.asp>, <http://web.developpez.com/tutoriel/xhtmlcss/>, ...

## 8 Editeurs HTML

Nous avons vu comment fonctionnait une page html. On peut légitimement penser que c'est quand même laborieux et qu'il serait pratique de disposer d'un logiciel permettant directement d'éditer une page comme on désire la voir s'afficher. Quand on édite un document avec Word, c'est lui qui se charge des styles, ça nous est transparent et c'est bien pratique.

C'est la notion « tel-tel » : tel affichage, tel résultat. WYSIWYG en anglais, acronyme de la locution anglaise « What you see is what you get » signifiant littéralement « ce que vous voyez est ce que vous obtenez ». Cette notion est utilisée en informatique pour désigner les interfaces utilisateur graphiques permettant de composer visuellement le résultat voulu. Elle désigne, de fait, une interface « intuitive » : l'utilisateur voit directement à l'écran à quoi ressemblera le résultat final.

Un éditeur tel-tel permet d'éditer une page Web à peu près telle qu'elle apparaît dans les navigateurs courants. Il propose les fonctions classiques des traitements de texte tel-tel. Ses deux principaux avantages sont la facilité d'utilisation et l'observation immédiate du rendu graphique. Ses deux principaux désavantages sont le manque de maîtrise sur la qualité du document HTML produit et les risques d'incompatibilité avec des navigateurs non prévus par l'éditeur (plus récents, plus anciens, moins courants...) si celui-ci ne suit pas les recommandations en vigueur telles qu'édictées par le W3C.

Pourquoi ne pas avoir commencé par là ?

Parce qu'il est très souvent utile de comprendre le code généré par l'éditeur. Quand on n'arrive pas à générer ce qu'on veut :

- soit parce qu'il peut être plus compliqué de se servir de l'interface que de générer le code soit même,
- soit parce que la fonction que vous cherchez n'a pas été implémentée,
- soit parce qu'il y a tout simplement un bug dans l'éditeur.

Quels éditeurs sont disponibles ?

- Payants :
  - Adobe Dreamweaver
  - Microsoft FrontPage
  - Webexpert
  - ...
- Gratuits
  - Nvu
  - HTML Kit
  - ...

## Annexe 1 : Sources d'informations

Le web est la meilleure source de renseignements pour la recherche d'informations à propos du web :

- Une encyclopédie libre issue d'un effort de collaboration des internautes : [Wikipedia \(http://fr.wikipedia.org/wiki/Accueil\)](http://fr.wikipedia.org/wiki/Accueil).
- La [norme](http://www.la-grange.net/w3c/html4.01/cover.html) définissant HTML (<http://www.la-grange.net/w3c/html4.01/cover.html>).
- Quelques cours et tutoriels : <http://web.developpez.com/tutoriel/xhtmlcss/>
- Comment ça marche ? <http://www.commentcamarche.net/>

Tous les exemples de ce cours proviennent d'un cours en ligne : <http://www.creation-de-site.net/langages/html.php>

Livres : Beaucoup de livres traitent d'HTML et du développement de sites Web. Pas tous de bonne qualité ni faciles à utiliser. De plus en informatique le plus souvent, on ne lit pas un livre du début à la fin, on cherche la solution d'un problème qui vient de se présenter lors d'un développement quelconque. Il est important donc que le livre soit rédigé avec de bons sommaire, index, glossaire ... Il n'est possible de se rendre compte de la qualité du livre que en essayant justement quelques recherches.

Un point important d'une manière générale en informatique, et plus encore quand il s'agit d'internet, est la date de parution. Un livre trop vieux, soit sera obsolète, soit ne bénéficiera pas d'innovations utiles : il est inutile d'apprendre à laver le linge à la main quand les lave-linges existent ! Si l'on n'est pas accro aux nouvelles technologies, inutile non plus de foncer sur les livres les plus récents. Les informations qu'ils contiennent peuvent être trop pointues ou manquer de maturité. Une technique est bonne quand elle a été éprouvée et autant laisser faire ce travail par les autres si l'informatique n'est pour vous qu'un outil.

Quelques références tous disponibles à la BU de sciences :

- Cas pratiques, initiation aux langages HTML et XML de Cassagne et Rampnoux chez Ellipses. Petit et concis.
- Guide de l'utilisateur HTML 4 XML chez Micro application. Un peu vieux mais en couleur, plein d'exemples et de copies d'écran.
- HTML entraînez-vous pour maîtriser le code source chez Eni. Plein d'exercices.
- HTML et XHTML la référence de Musciano et Kennedy chez O'Reilly. Un pavé mais il y a tout, plus pour les experts.

Un site écrit en pur HTML : <http://www-cs-faculty.stanford.edu/~knuth/>

## Annexe 2 : Comparaison HTML-XML

Alors qu'HTML est un langage à part entière, son vocabulaire est figé (Balises imposées). Il ne concerne que la mise en page, pas la syntaxe ni la signification. XML peut être considéré comme un méta-langage (Balises non imposées => définies par le concepteur). Permettant de définir d'autres langages, permettant de décrire des protocoles.

Eléments de syntaxe XML :

- Toute balise ouvrante doit donner lieu à une balise fermante `<maBalise> Bonjour`  
`</maBalise>`
- Certaines balises peuvent être uniques => terminées par / : `<maBaliseUnique/>`
- Les espaces, en dehors du balisage, sont préservés
- Le nom des attributs sont écrits en minuscule
- Les valeurs d'attributs sont entre guillemets
- XML utilise entièrement UNICODE (norme ISO 10646)
- «Casse dépendant »(minuscules ≠majuscule)
- les données peuvent contenir presque tous les caractères UNICODE : cadre général pour définir des tables de caractères.

## Annexe 3 : Les scripts

- CGI : Common Gateway Interface, traduit interface de passerelle commune, est une appellation pour un programme, script exécuté par le serveur web (on dit généralement « côté serveur »), permettant d'envoyer au navigateur de l'internaute un code HTML créé automatiquement par le serveur (basé par exemple sur une autre application, telle qu'un système de gestion de base de données, d'où le nom de passerelle).  
Un des principaux intérêts de l'utilisation de CGI est la possibilité de fournir des pages dynamiques, c'est-à-dire des pages personnalisées selon un choix ou une saisie de l'utilisateur. L'application la plus fréquente de cette technique repose sur l'utilisation de formulaires HTML permettant à l'utilisateur de choisir ou de saisir des données, puis de cliquer sur un bouton de soumission du formulaire, envoyant alors les données du formulaire en paramètre du script CGI.
- PHP : PHP est un langage interprété permettant d'écrire un CGI. La syntaxe du langage provient de celles du langage C, du Perl et de Java. Ses principaux atouts sont :
  - Une grande communauté de développeurs partageant des centaines de milliers d'exemples de script PHP ;
  - La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL) ;
  - La simplicité d'écriture de scripts ;
  - La possibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGI, pour lesquels il faut écrire des lignes de code pour afficher chaque ligne en langage HTML) ;
  - La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur de nombreuses plateformes : Unix, Linux, Windows, MacOS X, Solaris, etc...) ;
  - L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, etc.).
- ASP : Active Server Pages est un « concurrent » de PHP, standard mis au point par Microsoft en 1996
- Javascript : Le Javascript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.  
Ainsi le langage Javascript est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage Java, avec lequel il a longtemps été confondu.
- Applet Java : Les applets sont de petites applications graphiques destinées à s'exécuter dans un navigateur internet, donc côté client. D'autre part, les applets ont un champ d'action limité pour des raisons de sécurité puisqu'elles s'exécutent dans un environnement distant. Elles ne peuvent donc pas accéder à toutes les ressources de l'environnement distant (lecture/écriture sur le disque par exemple).

## Annexe 4 : Résumé des balises

Nom	Fonction
<!-- -->	Permet de placer dans le code source d'une page des commentaires qui ne seront pas affichés dans le navigateur.
A	Crée un lien hypertexte vers une autre page web ou vers un ancre.
B	Pour avoir du texte en gras.
BIG	Augmente légèrement la taille du texte.
BLOCKQUOTE	Applique un retrait à un texte.
BODY	Définit le corps (contenu affichable) de votre page web. Il doit venir juste après l'en-tête de la page HEAD.
BR	Ajoute un saut de ligne.
CENTER	Centre le texte.
DIV	Regroupe des éléments pour en contrôler la structure avec des feuilles de styles par exemple.
FONT	Permet de mettre de modifier du texte (couleur, taille, police,...).
FORM	Permet de créer un formulaire.
H1 -> H6	Définit 6 niveaux pour les titres. Chaque niveau possède son propre style.
HEAD	Définit l'en-tête de votre document. Ces informations ne seront pas affichées par le navigateur.
HR	Place une séparateur horizontal.
HTML	Indique que le fichier est un document HTML.
I	Met le texte en italique.
IMG	Ajoute une image dans votre page.
INPUT	Ajoute un champ de formulaire (plusieurs types sont disponibles).
LI	Crée un nouvel élément pour une liste.
LINK	Permet de créer un lien vers un fichier externe (une feuille de style par exemple).
MARQUEE	Permet de faire défiler un texte (IE uniquement).
META	Définit des paramètres supplémentaires dans l'en-tête de votre page (très utile pour le référencement).
OL	Crée une liste numérotée.
OPTION	Crée un nouveau choix dans un SELECT.
P	Constitue un nouveau paragraphe.
SELECT	Crée une liste de sélection dans un formulaire.
SMALL	Diminue sensiblement la taille du texte (au contraire de BIG).
SPAN	Facilite la mise en forme (similaire à DIV pour des portions de textes plus courtes).
STRONG	Affiche le texte en gras (similaire à B).
STYLE	Permet de définir des règles de style à utiliser sur le document.
TABLE	Crée un tableau.
TD	Crée une nouvelle cellule dans un tableau.
TEXTAREA	Crée une zone de texte dans un formulaire.
TITLE	Détermine le titre qui sera affiché dans la barre de titre du navigateur. Cette balise doit venir dans l'en-tête (HEAD) de votre code
TR	Crée une nouvelle ligne dans un tableau pour y insérer des cellules (TD).
U	Affiche le texte en souligné.
UL	Crée une liste à puces.