Le Langage HTML

Une introduction

Nino

Janvier 1997

SOMMAIRE

I - Introduction

Le but de ce document est de présenter, d'une façon concise et rapide, le langage HTML. Il ne s'agit que d'une introduction, les notions avancées comme les pages dynamiques, les extensions propriétaires des fournisseurs de navigateurs web ainsi que les langages comme Java et Javascript ne seront qu'évoquées dans le dernier chapitre, sans être développées.

Le lecteur est supposé connaître les principes d'Internet et savoir utiliser des outils classiques comme les navigateurs web.

Ce document n'a nullement la prétention d'être un cours sur le langage HTML. De plus, l'auteur ne peut assurer le lecteur de la totale exactitude des informations contenues dans les lignes qui suivent.

II - Rappels et définitions

Avant d'aborder le langage HTML, rappelons quelques définitions et concepts liés aux serveurs web.

Il existe, sur Internet (et maintenant en *Intranet*[1]), un nombre important de serveurs proposant des documents *multimédia* c'est-à-dire composés de texte, images, sons, animations etc. Ces documents, appelés *pages web* sont visualisés grâce à un programme appelé *browser* ou navigateur en Français. Il existe principalement deux grands navigateurs, à savoir Netscape, disponible sur un très grand nombre de plates-formes (Windows, UNIX, Mac) et Internet Explorer, fourni par Microsoft pour ses systèmes d'exploitation.

Un document, proposé par un serveur web, est toujours désigné par une "adresse" appelée *URL* pour *Universal Resource Locator*. Une URL[2] a la structure suivante :

protocole://adresse_machine[/répertoire/sous-rep/.../nom_document][3]

Le protocole *HTTP (HyperText Transfert Protocol)* est utilisé pour transmettre une page au format HTML (que nous allons définir tout de suite). Il est cependant possible d'indiquer un autre protocole comme ftp, telnet etc. C'est pourquoi on parle de localisateur de ressource **universel**.

La machine sur laquelle fonctionne le serveur web[4] est spécifiée soit par son adresse IP numérique (ex : 147.25.48.54) soit par son adresse DNS alphanumérique (ex : www.ibm.com). Dans le dernier cas, le premier terme de l'adresse DNS est **généralement** "www" (mais ce n'est **pas** une obligation).

Enfin, le document demandé est identifié par son chemin d'accès (*path*) et son nom. Pour une page HTML, son nom se termine par **.html**[5]. Notez que si aucun document n'est spécifié, c'est une page par défaut, dite *HomePage* ou Page d'acceuil qui est envoyée au navigateur.

Voici un exemple d'URL (fictive):
http://www.apple.com/index/download/drivers/list.html

Précisons que les pages HTML utilisent la technique de l'hypertexte, permettant de spécifier des URL au sein des pages, vers d'autres pages. Cela crée une structure, au niveau mondial, qui peut être vue comme une toile d'araignée, d'où le terme de *World Wide Web*.

[1] Voir le dernier chapitre.
[2] Bien que *locator* soit masculin, il est fréquent de parler d'*une URL* en Français.
[3] Les crochets indiquent une zone **optionnelle**. Il ne faut jamais les indiquer dans une URL.
[4] Ou FTP, Telnet etc.
[5] Parfois, on rencontre des pages avec l'extension .shtml (ce sont des pages dynamiques, Cf. dernier chapitre).

Maintenant que nous avons fait ces quelques rappels, il est temps de définir ce qu'est le langage HTML.

HTML signifie *HyperText Markup Language*. Comme son nom l'indique, c'est un langage qui permet de définir l'habillage d'un document, c'est à dire la façon dont il doit s'afficher à l'écran d'un navigateur.

Cette notion d'habillage est importante : elle signifie qu'une page écrite en HTML comportera du texte, bien-sûr, mais aussi des codes ou **balises** permettant de modifier l'affichage de ce texte, à savoir sa forme, sa taille, sa couleur. Le HTML permet également d'inclure des images, du son ou des animations dans une page web.

Le HTML est très simple. Ce n'est **pas** un langage de programmation dans le sens où il n'existe pas de variables, boucles, expressions conditionnelles. En fait, c'est plus un ensemble de codes qu'un langage, comme on le conçoit en informatique[6].

Il faut également signaler qu'un document HTML n'est autre qu'un **fichier texte** (i.e. ASCII) auquel on a ajouté des balises HTML. Il est en effet parfaitement possible de créer une page HTML avec un simple éditeur de texte comme Write ou Wordpad sous Windows, à condition de sauver le fichier ainsi créé au format texte, avec une extension .html (ou .htm pour les versions de Windows antérieures à Windows 95).

En pratique cependant, on utilise des outils munis d'une interface graphique évoluée permettant, d'un clic de souris, d'insérer dans le document en cours d'édition une balise HTML, comme on le ferait avec un traitement de texte.[7]

Dans les lignes qui suivent, nous n'allons présenter que **certaines** commandes du langage HTML, dans sa version 2, dernière en date à avoir été "officialisée". Nous ne présenterons que très peu de commandes appartenant aux dernières version du langage, toujours non définies officiellement, désignées par les numéros de versions 3 et 3.2. En pratique, les navigateurs récents exploitent déjà les fonctionnalités des versions 3 et 3.2 du langage.

[6] Un peu comme le *vidéotex* utilisé par le Minitel. Nous verrons, dans le dernier chapitre, que le HTML peut être associé à un ou plusieurs programmes.

[7] Dans le cas d'un logiciel de traitement de texte, comme Word, les caractères permettant de changer les attributs d'un texte (taille, corps, etc.) n'apparaissent pas dans le document mais ils existent bien et sont stockés lors de l'enregistrement du fichier.

III - Structure d'une page HTML

Nous avons vu dans le chapitre précédent qu'une page HTML était en fait un fichier texte, enrichi d'un certain nombre de codes ou commandes, appelées balises.
Ces balises sont toujours exprimées sous la forme d'un mot clé, encadré par les caractères "<" et ">". Exemple : <BALISE>.
Pour la plupart des balises, il existe une balise de fermeture associée, reprenant le même nom, mais précédé du caractère "/". Exemple : </BALISE>. La commande spécifiée s'applique donc **uniquement** au texte situé entre le couple de balises ainsi formé.
Exemple:
<HTML>
...
</HTML>

Notons que :

- une balise peut indifféremment être indiquée en minuscules ou en majuscules,
- le formattage "manuel" du document (espaces, sauts de lignes,...) est toujours ignoré.

Par exemple : **<HTML>**...**</HTML>** est interprété de la **même façon** par le navigateur web que la syntaxe sur plusieurs lignes indiquée ci-dessus.

Nous venons de découvrir, sans le savoir, un des éléments de la structure d'une page HTML : toute page doit en effet débuter par la balise **<HTML>** et se finir par **</HTML>**.

Entre ces deux balises, on définit deux zones : l'en-tête, spécifié par les commandes **<HEAD>** et **</HEAD>**, ainsi que le corps, délimité par: **<BODY>** et **</BODY>**. Ce qui donne, comme structure de base :

<HTML>
<HEAD>
...
</HEAD>
<BODY>
...
</BODY>
</HTML>

Dans l'en-tête, on ne met généralement qu'une seule information, le titre du document qui sera affiché en haut de la fenêtre du navigateur et qui apparaît dans les *bookmarks* (listes d'URL gérées par un navigateur, une sorte d'annuaire).
Ce titre est indiqué entre les balises **<TITLE>** et **</TITLE>**.

Exemple: **<TITLE>**Ceci est le titre**</TITLE>**.

Dans le corps, on met en fait tout le document à afficher (texte, définition des images etc.).

Signalons l'existence d'une balise de commentaire, qui peut être utilisée partout dans les documents HTML, définie comme suit :
<!--- Ceci est un commentaire **--->**.
Les commentaires ne sont jamais affichés à l'écran du navigateur.

En résumé, la structure générale d'une page HTML est la suivante.

```
<HTML>
<HEAD>
    <TITLE>Titre du document</TITLE>
</HEAD>
<BODY>
...
...
</BODY>
</HTML>
```

Notez que l'indentation facilite la lecture du code mais n'est pas reproduite par le *browser*.

IV - Styles d'écriture et séparateurs

Les principaux styles gérés en HTML sont le soulignage, l'*italique* et le **gras**. Ces modificateurs de styles sont définis par les commandes respectives **<U>...</U>**, **<I>...</I>** et **<B>...</B>**.

Par exemple, le document contenant :
<U>Souligné**</U>, <I>**Italic**</I>** et **<B>**Gras**</B>** s'affichera à l'écran du navigateur comme suit :
Souligné, *Italic* et **Gras**.

Il est également possible de changer la taille de la police de caractères, afin de mettre en valeur des titres, sous-titres etc. Pour ce faire, on utilise la balise **<H***n***>...</H***n***>** où *n* peut prendre une valeur entière comprise entre 1 et 6 (dans l'ordre **décroissant** de taille>.

En ce qui concerne les commandes de type séparateur, le HTML donne la possibilité de :
- revenir à la ligne[8] : **
**
- définir un paragraphe : **<P>**
- afficher une ligne horizontale : **<HR>**.

Vous remarquerez que les séparateurs ne fonctionnent pas par paire, contrairement à la majorité des commandes HTML.

[8] Rappelez-vous que des retours à la ligne dans le texte lui-même ne sont pas reproduits lors de l'affichage. D'où le besoin de cette balise.

V - Listes d'éléments

Le langage HTML permet de définir deux styles de listes : les listes numérotées et les listes non numérotées.

Une liste numérotée est délimitée par les balises **<OL>** et **</OL>** entre lesquelles chaque élément de la liste est précédé de la balise **<LI>**.

Exemple :
<OL>
    <LI>élément 1,
    <LI>élément 2,
    <LI>élément 3.
</OL>

Ce qui donne à l'écran :
1. élément 1,
2. élément 2,
3. élément 3.

Une liste non numérotée est définie de la même façon, sauf qu'elle est encadrée par les balises **<UL>** et **</UL>**.

Exemple :
<UL>
    <LI>élément 1,
    <LI>élément 2,
    <LI>élément 3.
</UL>

On obtient cette fois :
- élément 1,
- élément 2,
- élément 3.

Notez qu'il est tout-à-fait possible d'imbriquer des listes du même type ou de types différents.

Exemple :

<UL>
    <LI>élément 1 :
    <OL>
        <LI>sous-élément A,
        <LI>sous-élément B.
    </OL>
    <LI>élément 2,
    <LI>élément 3.
</UL>

On obtient alors :

- élément 1,
  1. sous-élément A,
  2. sous-élément B.
- élément 2,
- élément 3.

Enfin, il est possible de définir des listes spéciales, permettant de faire des glossaires, en utilisant les balises suivantes : **<DL>** et **</DL>** pour encadrer la liste, **<DT>** pour spécifier un élément et **<DD>** pour indiquer sa définition.

Exemple :
<DL>
    <DT>HTML**<DD>**HyperText Markup Language
    <DT>HTTP**<DD>**HyperText Transfert Protocol
</DL>

Donne :

HTML
    HyperText Markup Language
HTTP
    HyperText Transfert Protocol

VI - Liens internes et externes

En HTML, il existe deux types de *liens* : ceux qui sont internes à un document c'est-à-dire qu'ils désignent un point précis, repéré par une balise, dans le corps même du document, et ceux qui spécifient simplement une autre page HTML, un document multimédia ou l'accès à un service comme FTP. Nous allons voir les commandes utilisées pour définir ces liens.

On définit le point vers lequel pointe un lien interne en spécifiant une étiquette, affectée à une portion de texte, avec les balises suivantes :
<A NAME="nom_étiquette">...**</A>**.

Pour indiquer un lien vers cette étiquette, on utilise la syntaxe :
<A HREF="#nom_étiquette">...**</A>.**

Prenons un exemple afin de clarifier les choses :

Cliquer ici pour obtenir la définition du terme **<A HREF="#def">** HTTP**</A>
**
**
**
<DL>
<DT>HTML**<DD>**HyperText Markup Language
<A NAME="def"><DT>HTTP**<DD>**HyperText Transfert Protocol**</A>**
</DL>

On obtiendra alors dans la fenêtre du navigateur :

Cliquer ici pour obtenir la définition du terme HTTP.

HTML
HyperText Markup Language
HTTP
HyperText Transfert Protocol

Notez que le mot "HTTP", situé dans le code HTML entre les deux balises **<A HREF...>** et **</A>** est mis en évidence avec un soulignement (et un changement de couleur en pratique). En cliquant dessus, le navigateur va positionner le texte à la ligne où figure l'étiquette pointée par le lien, à savoir ici le mot "HTTP". Il faut également savoir que la zone encadrée par les balises **<A NAME...>** et **</A>** définissant l'étiquette n'est pas mise en évidence et n'est pas cliquable.

Considérons maintenant le cas d'un lien externe. Il suffit en fait de reprendre la syntaxe rencontrée ci-dessus pour définir un lien, en remplaçant le label par une URL, soit : **<A HREF="URL">**...**</A>**.

Exemple :

Cliquez **<A HREF="http://www.ibm.com">**ici**</A>** pour accéder au serveur web d'IBM.

A l'affichage on obtient :

Cliquez ici pour accéder au serveur web d'IBM.

En cliquant sur le mot "ici", le navigateur va se connecter au serveur web dont on a indiqué l'URL et afficher la page d'acceuil du serveur.

Comme nous l'avons dit au début de ce paragraphe, on peut aussi créer un lien vers un fichier multimédia. Prenons l'exemple d'une image GIF.

Il est possible d'**<A HREF="http://www.meteo.fr/cartes/lundi_1200.gif>**afficher **</A>** la carte du ciel de Lundi dernier à 12H.

Donne :

Il est possible d'afficher la carte du ciel de Lundi dernier à 12H.

En cliquant sur "afficher", l'image (fictive) ainsi définie sera affichée à l'écran.

Certains types de fichiers, comme les animations, ne peuvent être traités que grâce à un utilitaire externe au navigateur, pouvant être lancé automatiquement dès la fin du transfer du fichier (en examinant son extension).

Enfin, on peut créer un lien vers un serveur FTP grâce à une syntaxe du type : **<A HREF="ftp://ftp.sun.com/pub">**Répertoire public du serveur FTP de Sun**</A>.** Dans ce cas, le navigateur affichera à l'écran le contenu du répertoire ainsi défini.
Citons également le service mailto:adresse_email, qui peut être utilisé comme adresse de destination d'un lien. Dans ce cas, le fait de cliquer sur le mot désignant ce lien affiche à l'écran une fenêtre permettant d'envoyer un E-MAIL à l'adresse indiquée.
Exemple : **<A HREF="mailto:president@whitehouse.gov">**Ecrire au président des Etats Unis**</A>**.

Nous verrons dans le chapitre suivant comment définir un lien accessible en cliquant sur une image.

VII - Inclusion d'images

Dans le chapitre précédent, nous avons décrit comment définir un lien vers une page HTML ou une image. Dans ce dernier cas, l'image ainsi désignée s'affiche seule à l'écran du navigateur, sans qu'il ne soit possible d'y associer du texte, dans la même page. Nous allons donc voir comment insérer une image **dans** une page HTML.

Pour ce faire, on utilise la balise dont la syntaxe est la suivante :
<IMG SRC=/rep/sous-rep/.../image.gif>.
On peut également faire référence à une image se trouvant sur un **autre** serveur en indiquant son URL.
Exemple : **<IMG SRC="http://www.amiga.de/images/ami_logo.gif"**>.
Mais attention, ici il ne s'agit pas d'un lien mais toujours d'une inclusion. La seule différence par rapport à la syntaxe précédente est que dans le dernier cas, l'image ne se trouve pas sur le même serveur que la page qui y fait appel : le navigateur doit donc aller chercher cette image avant de pouvoir l'afficher.

La balise **<IMG SRC...>** possède une option permettant de spécifier une contrainte d'alignement par rapport au texte de la ligne où l'on désire insérer l'image. Cet argument est défini par le mot clé **ALIGN=mode**, où mode correspond à **TOP** (alignement sur le haut de l'image), **MIDDLE** (sur le milieu) et **BOTTOM** (bas).

Exemple :
Voici le logo de Netscape : **<IMG SRC="images/ns_logo.gif" ALIGN=BOTTOM>**

Voyons maintenant comment définir un lien auquel on accède en cliquant sur une image plutôt que sur une portion de texte. En fait, c'est très simple : il suffit de remplacer le texte définissant le lien par la balise d'inclusion d'une image.

Exemple:
<A HREF="http://www.netscape.com" ><IMG SRC=/images/ns_logo.gif></A>.

Dans cet exemple, le fait de cliquer sur l'image affiche la HomePage du serveur de Netscape. Notez qu'une image définissant un lien est différenciée des autres images par le fait qu'elle est entourée d'un cadre et que le curseur de la souris change de forme lorsqu'on le positionne sur cette image.

VIII - Concepts avancés

Dans les pages précédentes, nous n'avons présenté qu'une petite partie des balises HTML reconnues par les navigateurs les plus récents. Ce dernier chapitre est à considérer comme un catalogue non exhaustif de ce qu'il est possible de faire en HTML 3.2, sans préciser la syntaxe des balises utilisées. De plus, nous aborderons certains concepts liés à la programmation.

Le HTML 3 ajoute des possibilités à certaines commandes, et en apporte des nouvelles. Par exemple, il est possible d'ajouter des options à la balise **<BODY>** afin de spécifier la couleur des caractères, des liens, voire même une image de fond.

On peut également créer des tableaux contenant toutes sortes d'objets, comme du texte, des images, des liens etc. De même, il est possible de créer des formulaires de saisie (nous en reparlerons plus loin).

Citons enfin une fonctionnalité qui est désormais de plus en plus utilisée en HTML, il s'agit des *frames*. Les frames permettent de partager l'écran du navigateur en plusieurs fenêtres indépendantes, pouvant posséder certains attributs comme le redimensionnement ou la possibilité de faire défiler le contenu de la frame grâce à un ascenseur. L'application la plus classique est le partage de l'écran en deux frames, l'une contenant le sommaire d'un serveur (i.e. une liste de liens vers des pages correspondant aux principaux thèmes abordés par le serveur) et l'autre permettant d'afficher les pages sélectionnées via le sommaire.

Nous avons vu jusqu'à maintenant comment définir une page HTML statique, c'est-à-dire qui donne toujours le même résultat lors de son chargement à l'écran du navigateur. Cependant, dans certains cas, il peut être nécessaire d'utiliser des pages dynamiques, créées en fonction de requêtes émises par l'utilisateur, via un formulaire par exemple. Dans ce cas, on doit recourir à ce qu'on appelle des scripts ou programmes *CGI (Common Gateway Interface)*.

Un script ou programme CGI peut être écrit soit avec un langage de type shell, soit avec un véritable langage de programmation comme le C ou le Perl. Dans tous les cas, le serveur reçoit une requête de la part du navigateur, à partir de laquelle le programme CGI va effectuer le traitement demandé et produire, en fonction des résultats de ce traitement, une page HTML dynamique, envoyée au navigateur exactement comme le serait une page statique. A noter que le script ou programme CGI est exécuté **sur le serveur**.

Depuis peu, il est possible d'utiliser un langage appelé *Java* afin de construire une application directement exécutée **sur le poste client**, via le navigateur. Pour ce faire, le serveur envoie au navigateur ce qu'on appelle une *applet* afin que celui-ci

l'exécute. Cette applet peut être un véritable programme, avec interface graphique. Cette technique permet de distribuer la charge de travaille afin d'alléger les serveurs HTTP.
Java est un langage à part entière, orienté objet et proche du C(++), mais non compilé. Nous n'en dirons pas plus en ce qui le concerne, tellement le sujet est vaste...

Nous finirons ce chapitre en disant quelques mots sur ce qu'on appelle l'*Intranet*. Ce concept est apparu il y a quelques années, lorsque les entreprises ont découvert l'intérêt des technologies liées à Internet. Un Intranet, c'est tout simplement un réseau (local ou étendu), propre à un organisme (entreprise, administration,...) qui utilise les techniques de l'Internet (principalement celles liées au web) afin de fournir un service unifié au personnel de l'organisme, basé sur le navigateur. On peut citer comme applications la consultation de bases de données internes (stocks, commandes,...), l'accès à des données techniques propres à l'organisme (normes,...), des services d'information etc. Cette fois encore, nous pourrions en dire plus, mais cela sortirait du cadre de ce document.

IX - Conclusion

Ce document vous a donc présenté un certain nombre de concepts de base concernant le langage HTML. Comme son titre le précise, il ne s'agissait que d'une introduction, volontairement succincte. Si vous voulez en savoir plus, sachez qu'il existe des dizaines d'ouvrages consacrés à ce sujet, en Français ou en Anglais, qui détaillent parfois l'intégralité du langage dans ses moindres subtilités, exemples à l'appui.

Si vous avez l'intention de concevoir une page web, il est conseillé de commencer par rechercher un outil d'aide à la conception. Il en existe beaucoup dans le domaine public, citons par exemple HotMetal, WebExpert ou HotDog, réputés pour leur puissance et/ou leur simplicité d'utilisation. Notez qu'il existe également certains outils dits *WYSIWYG (What You See Is What You Get)* qui permettent de créer des pages HTML simples, sans avoir besoin d'entrer la moindre commande à la main (il en résulte bien-sûr une certaine limitation dans la liberté de conception). Microsoft propose aussi une extension à son célèbre Word, permettant de créer des pages HTML.

N'oubliez pas, lorsque vous rencontrez une page web qui vous intrigue, que vous pouvez afficher son code source grâce à une commande de votre navigateur (Menu "Affichage/Source du document" sous Netscape), afin de l'étudier.

."