Le langage HTML du WWW.

Emmanuel PIERRE epierre@mail.esiea.fr.

Internet a rendu populaire son "langage", le scripte appelé HTML(Hyper Text Markup Langage), ‘markup' pour les étiquettes intégrées dans la page. C'est un scripte initialement développé par le CERN, qui voulait obtenir un ensemble de documents liés par leurs adresses, et permettant de s'entre appeler suivant les références désirées. Ce langage a évolué, depuis sa première spécification qui ne permettait pas d'interactivité, jusqu'à la troisième spécification qui permet de remplir des formulaires, cliquer sur des images pour obtenir plus de détails...

Aujourd'hui il est devenu le média incontournable de tout ce qui tourne sur Internet et les W3 (réseaux basés World Wide Web), les pages personnelles et les présentations diverses.

Cet article introduira la norme initiale, puis ses évolutions via HTML+ (les formulaires, boites de choix), et enfin les liens sur images, et introduira des méthodes pour traiter les informations reçues de ces formulaires.

0. Introduction à la HTML

Les commandes sont délimitées par < et >. Pour annuler un effet actif, le slash (/) devant le nom d'effet le termine : </*commande*>. Les commandes sont des abréviations mnémotechniques des noms anglais des mises en forme comme <p> pour un saut de ligne. Il est à noter qu'il n'y a un saut de ligne que si <p> est explicitement mis (Netscape ou Mosaic mettant en forme les paragraphes automatiquement) :

ex: ceci est un
test

donnera : ceci est un test.

Ex: ceci est un texte qui s'allonge<p>et repart sur une nouvelle ligne.

Après cet exemple, voyons le corps d'un scripte HTML. Il commence par un entête <HTML>, suivi de la commande <HEAD>, qui inclut le titre <TITLE> qui va apparaître dans la barre au dessus du visualisateur. Le titre est un attribut persistant, il faut donc l 'annuler juste après: </TITLE>, comme la fin de l'entête </HEAD>, pour commencer le corps de la page <BODY>. Entre le début du corps et la fin du corps se trouvera l'essentiel de la page, ses markups et ses informations. Pour mémoire, voici le prototype du corps d'une page ci-après :

```
<HTML>
```

```
<HEAD>

<TITLE> </TITLE>

</HEAD>

<BODY>

</BODY>

</HTML>
```

En gros, le fichier HTML ne comprend que des commandes, la mise en page, et le reste sera fait par le browser. Il existe d'ailleurs un grand nombre d'effets sur le texte dans une page.

C'est dans la directive <BODY> que se définit la texture de fond (<BODY BACKGROUND="http://monserveur/monimage">), ou la couleur des liens (LINK), des liens déjà visités (VLINK) et du texte (TEXT). Cela se fait en définissant la palette RGB e.g. L INK=#FF0000 définit le taux de rouge en tant que FF, et le reste nul.

1. Les entêtes :

Il existe 6 catégories de titres, depuis H1 le plus grand, jusqu'à H6 le plus petit. Ce sont là aussi des attributs persistants, donc ne pas oublier le bloc de terminaison :

```
<H1></H1>

<H2></H2>

<H3></H3>

<H4></H4>

<H5></H5>

<H6></H6>
```

2. Format général

Le format général représente les commandes les plus communes pour la mise en page du document.

Le titre du document, comme déjà vu plus haut, fait parti du format général, les deux commandes spécifient le titre à afficher sur la barre de statut de la fenêtre active. Commande : <TITLE>...</TITLE>

Pour séparer les blocs de caractères, il est souvent utile de laisser quelques lignes, et un séparateur, cette commande existe et c'est <HR>, qui trace une ligne horizontale (Horizontal Line), c'est un attribut local (ce n'est pas la peine de le désactiver) : <HR> .

De la même façon, comme mentionné plus haut, le texte ne fait pas de retour à la ligne automatique, seulement quand un <P> (new Paragraph) est présent. Il existe aussi une commande <BR>,(line Break) qui coupe la ligne de texte et va à la ligne. Pour finir dans le général, il faut citer 3 commandes de style:

<pre>...</pre> : texte préformatté.

<listing>...</listing> : style listing, fonte appropriée.

<blockquote>...</blockquote> : citation, à la ligne et centré.

Par ailleurs, pour écrire en bon français, il faut tenir compte que les fichiers http sont sur 7 bits, et donc il faut coder les caractéres nationaux:

È pour È

é pour é

Ç pour Ç

...

pour en savoir plus, lire http://web/urec.fr/docs/WWW/ISOlat1.html

3.Styles logiques

C'est une suite de commandes qui agiront sur l'affichage du texte concerné, comme par exemple la taille, le style ... En général, ce sont des styles les plus représentatifs du style de texte à afficher :

... : Emphase, insiste sur le texte concerné.

... : Emphase plus forte sur le texte.

<code>...</code> : Affiche une commande HTML.

<samp>...</samp> : Affiche en style sortie texte sous DOS.

<kbd>...</kbd> : Affiche une touche clavier.

<var>...</var> : Définit une variable (à affichage).

<dfn>...</dfn> : Affiche une définition.

<cite>...</cite> : Affiche une citation.

<tt>...</cite> : Affichage mode texte.

Tous ces styles n'apparaissent pas différemment sur tous les NetBrowser, celui que j'utilise, Netscape, traite notamment samp et kbd de la même façon. L'intérêt est bien sûr la grande practicité de ces styles, mettant en relief de nombreux aspects.

4. Styles physiques

Les attributs sur le texte sont ceux d'un traitement de texte normal : **gras** (bold, lettre b), *italique* (italic, lettre i), souligné (underligned, lettre u). Pour l'instant seule la fonte par défaut est utilisable mais on envisage déjà de télécharger les fontes. Voici déjà les commandes sur le texte:

<u> </u>

On peut bien sur additionner ces attributs :

Bonjour le <u> monde </u>

Bonjour *le monde*

Un quatriéme style de caractére existe, <tt> qui affiche une fonte qui simule des caractéres de machine à écrire (TypeWriter Style).

5. Listes

Des commandes de liste existent et permettent d'organiser différents points, accompagnés de retraits, ainsi que de les numéroter.

Il existe deux types de listes, les listes ordonnées (Ordered Lists) qui pour chaque objet dans le bloc <ol>... désigné par <li>, objet de liste, donne un numéro (1, 2...), et impose un retrait, et les listes non ordonnées, (Unordered Lists) , qui dans le bloc <ul>.. met une puce avec un retrait du bord, pour chaque objet de la liste <li>.

Par exemple:

Voici 3 choix<p>
lire ce texte<p>
ne pas lire ce texte<p>
changer de cravate

cela donne :

Voici 3 choix

1. lire ce texte
2. ne pas lire ce texte

3. changer de cravate

Il existe aussi une commande pour afficher un menu, c'est <menu>...</menu>. En voici un exemple:

<menu>au menu ce soir<p>
carottes râpées<li>bœuf en daube avec garniture<li>îles flottantes</menu>

Pour des listes courtes (moins de 20 caractères), il existe une commande de répertoire, appelée <dir> :

ex:

<DIR>

<LI>A-H<LI>I-M

<LI>M-R<LI>S-Z

</DIR>

Des commandes existent aussi pour donner des définitions, encadrées par <d l> et </dl>, où <dt> désigne le terme à définir, et <dl> sa définition. La définition sera mise en retrait, à moins que l'on ait spécifié <dl compact> qui les mettra à la suite.

ex:

<DL>

<DT>Termes<DD>Ceci est la définition du premier terme.

<DT>Term<DD>Ceci est la définition du deuxième terme.

</DL>

6. Divers

Il existe bien-sûr une commande pour mettre des commentaires dans le texte d'une page HTML : <!- ... -> qui est d'un effet trés visuel, remplacez les ... par votre texte. N'en abusez pas, car le scripte HTML est transmis tel-quel avec les commentaires aux browsers, et donc en mettre trop augmente le volume de communications.

Dans la même rubrique, on peut définir des liens entre documents avec <link href="URL"[rev=][align=]> mais cela ne créera pas de lien hypertexte. On trouve aussi un formatage de texte <address></address> pour spécifier une adresse, celle-ci apparaîtra en i talique. On trouve aussi <base> qui affiche le chemin explicite de la page actuelle, ou <isindex> qui informe que l'on peut faire des recherche d'index sur la page courante.

7. Hyperliens et Ancrages.

L'attribut <A> permet des ancrages de documents, c'est à dire permet de créer des liens pour passer d'un lien à un autre, un lien pouvant être dans le même document, ou dans un autre.

Par exemple : <A HREF="http://www.esiea.fr/public_html/Emmanuel.PIERRE">voir Emmanuel PIERRE pour plus de détails</A> est un lien explicite vers une autre page dont la référence Hypertexte est définie par HREF="". Pour l'utilisateur verra seulement voir Emmanuel PIERRE pour plus de détails souligné et en bleu, ce qui lui signifiera le passage à un autre lien.

Mais ce lien peut aussi être dans le même document: <A HREF="document.html#glossaire">glossaire</A> définit un lien dans le même document, qui lui sera défini par <A NAME="glossaire">Glossaire</A> plus loin dans le même document (appelé localement ici docu ment.html). le nom (NAME) doit être unique dans le document. ex:

<A NAME=cafe>Caf,</A> est un exemple ...

Un exemple en est <A HREF=#cafe>caf,</A>.

8. Les URL (Uniform Ressource Locators).

Les URL sont des pointeurs vers des informations qui sont appelées quand un lien est activé par un utilisateur. Ils donnent au browser des informations sur la façon dont sont stockées les informations, et leur emplacement. Cela permet au browser WWW de reg rouper en lui-même des fonctionnalités qui normalement nécessiteraient plusieurs programmes différents, comme pour le téléchargement (FTP : File Transfer Protocol), les news, Email et gopher.

La forme générale d'une URL est :

`prefixe hote.domaine/chemin/fichier`

En voici quelques exemples :

ftp:// [télécharge un fichier d'un serveur FTP]

http:// [charge un fichier d'un serveur World Wide Web]

file://localhost/ [télécharge un fichier du même ordinateur que celui d'oú tourne le browser - un fichier "local"]

gopher:// [télécharge un fichier d'un serveur Gopher].

mailto: [demande au browser de transmettre un message Email] Emmanuel PIERRE

news: [charge un fichier des news Usenet] esiea.announce

Voici un exemple de commande correctement construite:

"ftp://ftp.cs.dal.ca/htmlasst/htmlasst.zip",

qui sert à télécharger dans un document HTML le fichier "htmlasst.zip", d'un site FTP.

"file:/users/a98/pierre/hpux/make-links",

fera afficher par le browser le fichier "make-links" du disque dur local de l'utilisateur.

"http://www.cfn.cs.dal.ca/Media/News/TodaysNews.html", chargera et affichera une page HTML d'un serveur WWW.

Dans un document, il est toujours préférable, lorsqu'on crée des liens vers d'autres pages, de les mettre de façon absolue, c'est à dire avec la location compléte chez l'hôte. On peut néanmoins faire des liens relatifs pour toutes ses pages en spécifiant < BASE HREF="http://www.esiea.fr/">, alors tous les liens tels que <A HREF="/public_html/Emanuel.PIERRE/index.html"> seront complétées en http://www.esiea.fr/public_html/Emmanuel.PIERRE/www.html

9. Les images.

Les éléments image sont utilisées pour incorporer en-ligne des images dans un document HTML. La syntaxe de base est <IMG SRC="FICHIER_IMAGE.GIF" ALT="texte">, le plus souvent et historiquement, le format de l'image est GIF (General Interlaced File), mais c omme une image peut être trés grande, ou que le terminal n'affiche pas en mode graphique (certains utilisent un browser en mode texte appelé lynx), le paramétre ALT sert à spécifier un nom alternatif, ce sera le texte affiché à la place de l'image. Il est aussi conseillé de spécifier la taille de l'image avec WIDTH=largeur_en_pixels, et HEIGHT=hauteur_en_pixels, ce qui permettera au browser de préparer un cadre pendant que l'image se charge, et donc de savoir comment présenter la page. Cela peut représenter un serieux gain de vitesse.

Le texte suivant de l'image peut être présenté de différentes façons, par le paramétre ALIGN, qui propose trois méthodes: TOP, MIDDLE, BOTTOM, et place l'image au-dessus, au milieu et en bas du texte respectivement.

Récemment, quatre nouveaux attributs ont été ajoutés pour améliorer la présentation image/texte:

ALIGN=TEXTTOP : aligne le sommet de l'image avec le sommet du texte.

ALIGN=ABSMIDDLE : aligne le milieu de l'image avec le milieu de la ligne, qui peut

contenir d'autres images.

ALIGN=BASELINE : comme BOTTOM.

ALIGN=ABSBOTTOM : aligne le bas de l'image avec l'objet le plus bas dans la ligne.

D'autres attributs ALIGN permettent de définir le placement de l'image par rapport au texte : ALIGN=LEFT, ALIGN=RIGHT qui placent respectivement l'image à gauche et a droite.

Pour affiner la présentation, il est souvent nécessaire de laisser de la place au dessus ou au desous de l'image. Cela se fait avec <BR CLEAR=attribut>, oú attribut est LEFT, RIGHT ou ALL. Cela damande de recommencer à placer du texte oú la marge à gauche, droite ou les deux marges sont vides.

On peut aussi demander au browser de laisser de la place entre les images et l'objet suivant (image ou texte) avec HSPACE=espace_horizontal_en_pixels et VSPACE=espace_vertical_en_pixels, et dont voici un exemple complet :

<IMG SRC="monimage.gif" ALT="un pinguin en short" WIDTH=57 HEIGHT=85 HSPACE=50 VSPACE=50 ALLIGN=ABSMIDDLE>

Pour mémoire, il existe encore un attribut pour les images, LOWSRC; quand le browser charge les images, il chargera d'abord celle spécifiée par LOWSRC, qui se charge rapidement (basse réolution ou plus petite taille), puis lentement et en tâche de fond ch arge celle spécifiée par SRC. C'est une façon d'augmenter la vitesse de chargement d'une image.

Une image comme un texte peuvent constituer un lien hypertexte :

```
<A HREF="http://machine/ht/info.html">

<IMG SRC="sample.gif">

</A>
```

L'image apparaîtra entourée d'un cadre bleu pour signifier qu'elle est un lien hypertexte, et cliquer dessus aiguillera vers la page info.html. Pour faire disparaître ce cadre, utilisez BORDER=0 dans l'<IMG SRC="" BORDER=0>.

En général si l'image est trés importante, il est conseillé de faire une icône ou une image plus petite liée à l'image entiére.

Le dernier attribut existant permet de cliquer sur une partie de l'image, et cette zone sera transmise à un programme serveur. L'attribut est ISMAP :

```
<A HREF="http://machine/cgi-bin/imagemap/sample">
<IMG SRC="sample.gif" ISMAP>
</A>
```

Le paramétre sera transmis en tant que http://machine/cgi-bin/imagemap/sample.map?coo_x,coo_y à l'application sample, qui traitera l'information et aiguillera en fonction de la réponse.

Certains serveurs permettent de faire toute la gestion de ISMAP à l'intérieur de la page, comme apache, auquel il suffit de donner le chemin du fichier .map comme s'il s'agissait d'une page html :

```
<A HREF="http://machine/sample.map">
<IMG SRC="sample.gif" ISMAP>
</A>
```

Que contient un fichier de MAP ? D'abord la forme (rect, poly, circle, ...) puis l'URL vers laquelle pointer, puis les coordonnées relatives au haut à gauche de l'image et x et y pixels éparés par une virgule. Par exemple pour une zone carrée, on aura l'e ntrée:

```
rect http://www.esiea.fr/eleves/index.html 50,50 100,100
```

Pour les formes polygonales, ne pas dépasser plus de 100 points.

10. Les tableaux.

Une table est définie par des entêtestes (Table Header), des lignes (Table Record) et des cellules (TD). On définit une table par le tag <TABLE> et se termine comme tout attribut par </TABLE>. On peut définir l'épaisseur du cadre du tableau avec BORDER=*valeur* dans le tag du tableau ainsi que la largeur avec l'attribut WIDTH=*valeur* qui peut être exprimé en pourcentage (%).

On peut ajouter un titre au tableau à travers le tag <CAPTION>Titre du tableau</CAPTION>.

Le tableau se définit de haut en bas, ligne par ligne suivant un <TR>. Pout définir les noms de colonnes on utilise <TH>nom</TH>, et pour les noms de ligne <TH ALIGN=left>nom</TH>.

Les tags TH et TD peuvent prendre plus de place que celle attribuée initialement en utilisant les attributs COLSPAN et ROWSPAN dans les tags, le premier accaparant le nombre de ligne demandé, et le second le nombre de colonnes. Voici un exemple:

```
<TABLE BORDER>
  <CAPTION>Test de table</CAPTION>
  <TR><TH ROWSPAN=2><TH COLSPAN=2>Moyenne
      <TH ROWSPAN=2>autre<BR>catégorie<TH>Divers
  <TR><TH>taille<TH>poids
  <TR><TH ALIGN=LEFT>pales<TD>1.9<TD>0.003
  <TR><TH ALIGN=LEFT ROWSPAN=2>hélices<TD>1.7<TD>0.002
</TABLE>
```

	Moyenne		autre catégorie	Divers
	taille	poids		
pales	1.9	0.003		
hélices	1.7	0.002		

Les attributs d'alignement doivent être définis dans chaque case. Dans la tag TABLE, l'attribut COLSPEC permet de féfinir des attributs pour les colonnes, par exemple COLSPEC="L20 C8 R13" définit que la première colonne est alignée à gauche (Left) et de taille 20, la deuxième centée (Center) et de 8, et la troisième alignée à droite de 13.

11. Les formulaires.

L'inclusion de formulaires dans les documents HTML permettant aux utilisateurs de rentrer des informations dans les documents est la grande différence entre la deuxiéme spécification du HTML et les précédentes. Il s'agit de champs d'entrée placés dans des paragraphes, dans du texte et même des listes. Cela confére aux documents une grande flexibilité. Voici les différents types :

<FORM> ... </FORM> : Un formulaire dans un document.
<INPUT ...> ... </INPUT> : Un champ d'entrée simple.
<OPTION> : Une option dans un formulaire.
<SELECT> ... <SELECT> : Une option par défaut dans un ensemble d'options.
<TEXTAREA ...> ... </TEXTAREA> : Un champ d'entrée sur plusieurs lignes.

Chaque variable de champ est définie par une entrée(Input), une zone de texte (Textarea), ou une option (Option) et doit avoir un nom pour être identifié dans les données transmises à l'application serveur. Voyons un exemple (se trouve à http://www.esiea.fr/esiea/inscr.html):

```
<H1>Demande de documentation<HR></H1>
<BR><BR>
<FORM METHOD=POST action="inscr.cgi">
<TABLE border=0 cellpadding=5 cellspacing=3 border=1 COLSPEC="L20 L20" >
<tr><td>Nom:</td>
```

<td><INPUT NAME="Nom" TYPE="TEXT" ROWS=1 COLS=1 SIZE="40"></td></tr>
<tr><td>Prénom:</td>
<td><INPUT NAME="prenom" TYPE="TEXT" ROWS=1 COLS=1 SIZE="40"></td></tr>
<tr><td>Date de naissance(JJ/MM/AA):</td>
<td><INPUT NAME="date" TYPE="TEXT" SIZE="10"></td></tr>
<tr><td>Adresse:</td>
<td><INPUT NAME="adresse1" TYPE="TEXT" ROWS=1 COLS=1 SIZE="60"></td></tr>
<tr><td></td><td><INPUT NAME="adresse2" TYPE="TEXT" ROWS=1 COLS=1 SIZE="60"></td></tr>
<tr><td>Téléphone:</td>
<td><INPUT NAME="tel" TYPE="TEXT" ROWS=1 COLS=1 SIZE="10"></td></tr>
<tr><td> </td><td> </td></tr>
<tr><td>Etudes précédemment faites:</td>
<td><INPUT NAME="etude" TYPE="TEXT" ROWS=1 COLS=1 SIZE="60"></td></tr>
</table>

<HR>
Choix de l'établissement (cocher la case correspondante):<P>
<table border=0 cellpadding=5 cellspacing=3 border=1 COLSPEC="L20 L20">
<tr><td>1er choix:</td></tr>
<tr><td>ESIEA Paris</td><td><INPUT NAME="premier" TYPE="RADIO" VALUE="Paris" ROWS=1 COLS=1 SIZE="40"></td>
<td>ESIEA Laval</td><td><INPUT NAME="premier" TYPE="RADIO" VALUE="Laval" ROWS=1 COLS=1 SIZE="40"></td></tr>
<tr><td>2ème choix:</td></tr>
<tr><td>ESIEA Paris</td><td><INPUT NAME="deuxieme" TYPE="RADIO" VALUE="Paris" ROWS=1 COLS=1 SIZE="40"></td>
<td>ESIEA Laval</td><td><INPUT NAME="deuxieme" TYPE="RADIO"
VALUE="Laval" ROWS=1 COLS=1 SIZE="40"></td></tr>
</table>
Que désirez vous?<P>
<table>
<tr><td align=right><INPUT NAME="brochure" TYPE="CHECKBOX" ALIGN=middle></td><td> Une brochure</td></tr>
<tr><td align=right><INPUT NAME="inscri" TYPE="CHECKBOX" ALIGN=middle></td><td> Un dossier d'inscription en 1ère année préparatoire</td></tr>
</table>

<table width=100%>
<tr><td><INPUT value="Transmettre les informations" TYPE="SUBMIT"></td>
<td align=middle><INPUT value="Effacer" TYPE="reset"></td></tr>
</table>
</FORM></H3>

Dans l'exemple, RESET réinitialise les formulaires, et SUBMIT transmet les informations. Il existe un grand nombre de méthodes pour entrer du texte, l'un de ceux-là est de permettre d'entrer du texte sur plusieurs lignes, comme avec le Textarea.

12. Recevoir des informations.

Quand le bouton SUBMIT est cliqué, la méthode appelée POST envoie une chaîne à l'application. Cette chaîne est une suite de *nom=valeur*, séparés par des &. Les espaces de nom et valeur sont remplacés par des signes + (plus). La méthode POST signifie que l' application recevra la chaîne sur l'entrée standard. Le protocole est défini par la norme Common Gateway Interface (CGI), qui prévoit aussi qu'une application peut répondre sur la sortie standard. (pour plus de détails :*http://hoohoo.ncsa.uiuc.edu/cgi/interface.html*).

```
void plustospace(char *str) {
register int x;
for (x=0;str[x];x++)
  if(str[x] =='+') str[x] = ' '
}
```

Cela transforme les '+' en espace. Un simple printf("Si vous n'avez pas compris,lisez");printf(" <A HREF=\"http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-forms/overvie.html\">Aspect général des formulaires </A>.%c",LF); et le tour est joué ! U ne nouvelle page s'affichera avec un lien hypertexte.

10. Scriptes CGI

C'est ce qui permet par exemple d'exécuter des commandes coté serveur et de récupérer le résultat pour le transmettre au butineur du client. Cela peut prendre la forme de programmes en C, comme le compteur de bas de page, ou de scripts en Perl, comme la recherche d'élèves, listes en tout genre.

L'avantage de Perl, est qu'il ait les mêmes fonctionnalités que le C, et ne demande ni allocation mémoire particulière, ni compilation. De plus la version actuelle 5.003 permet l'accès à des bases de données. Vous trouverez ici une courte introduction à perl très bien faite et un guide de référence et sa couverture (attention recto-verso).

Ces scriptes sont propres aux serveurs, ce sont des programmes qui sont exécutés dessus. De ce point de vue, ils peuvent être des trous de sécurité et pour cela, trés peu sont autorisés.

Néanmoins, ils sont le plus souvent réalisés en C (c.f. paragraphe précédent), ou en PERL, un Scripte indépendant de la machine serveur. Ce dernir permet des accés aux bases de données (Oracle, Ingres, PostGres95, mini-SQL...), mSQL permet même d'être utilisé à même le serveur httpd avec apache.

Les compteurs de pages sont trés appréciés par les connectés et les propriétaires de la pages. Ils permettent de savoir le succés de celle-ci. i le serveur n'en dispose pas, certains autres permettent gratuitement d'en avoir. Ils sont utilisables simplemen t comme des images et sont recaculés à chaque Reload de la page:

<IMG SRC="http://www.esiea.fr/cgi-bin/odometrer.cgi?page" ALIGN=ABSBOTTOM>

11. Question de style

On recommande en général de signer une page, c'est à dire d'y faire figurer son auteur, et des liens vers sa home page (page web personnelle) et un autre pour pouvoir lui écrire. Par exemple:

```
<HR> Dernière modification: 20 novembre 1996
<ADDRESS>
<A HREF="http://www.esiea.fr.fr/public_html/Emmanuel.PIERRE">Emmanuel PIERRE</A><BR>
<A HREF="mailto:epierre@mail.esiea.fr">epierre@mail.esiea.fr</A>
</ADDRESS>
```

Une autre façon de faire est de mettre ce lien dans l'entête:

```
<HEAD>
<TITLE>Tire de ma page</TITLE>
<LINK REV="fait par" HREF="mailto:epierre@mail.esiea.fr">
</HEAD>
```

13. HTML+ et HTML3

Pour n'en citer que quelques uns, on peut changer la taille de la fonte par <FONT SIZE=valeur> qui ira de 1 à 7, de défaut est 3, et aussi appliquer un changement à toutes les fontes de la page par <BASEFONT SIZE=valeur >. On peut aussi centrer son texte p ar <CENTER>, [... A COMPLETER]

13. Server Side Includes aka SSI

Les SSI sont là pour aider à personnaliser les pages. Jusqu'ici, l'exécution d'un CGI se faisait par liens HREF, celui-ci devant gérer tout l'affichage. Les SSI apportent l'altenative que le serveur va scanner la page à la recherche de commandes (habituellement contenues entre `<!--# -->`), inclus ou execute ces commandes, et inclus leur résultat dans la page, puis la transmet au buttineur.

Par exemple, je peux vous dire que vous lisez ces lignex avec Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt) depuis (none), que aujourd'hui est 28/01/2001, et qu'il est 21:10.

Plus en détail, pour:

- inclure un fichier, comme par exemple le bas de page ici, utiliser

  `<!--#include virtual="[/chemin absolu/]fichier.extension"-->`

- afficher une variable, comme par exemple le nom du document, utiliser

  `<!--#echo var="nom_de_variable"-->`, voir http://www.sigma.net/tdunn/ssi2/echo/echo.html pour plus de détails.

- exécuter un CGI, comme le compteur de page, utiliser

  `<!--#exec cgi="[/chemin absolu/]fichier.extension"-->`, ici on a appelé /cgi-bin/ref.cgi.

- exécuter une commande shell, utiliser

  `<!--#exec cmd="[/chemin absolu/]fichier.extension"-->`.

Pour une liste complète des variables d'environnement, cliquer ici.

Attention, on appelle ces commandes *Server* Side Includes car ces requètes sont traitées par le serveur. Ils ne fonctionneront donc pas si vous ouvrez des pages locales avec votre butineur, vous *devez* utiliser un serveur http pour cela.

Futur: Avec Apache 1.2 qui est encore en béta-tests, on peut inclure des expressions logiques dals les pages, par exemple tester si on a affaire à un netscape 2 ou non:

```
<!--#if expr="$HTTP_USER_AGENT=/Mozilla\/2/"-->
<!-- on fait ceci -->
<!--#else -->
<!--# sinon cela -->
<!--#endif -->
```

Exemples

Pour faciliter le référencement par des moteurs de recherche, on peut utiliser des ALTernatives texte aux images, en ajoutant ALT="mon texte" dans sa clause.

On peut aussi rajouter des information dans l'entête:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.2//FR">
<HTML>
<HEAD>
<TITLE> Mon titre </TITLE>
   <META NAME="GENERATOR" CONTENT="mon éditeur de texte [VI]">
   <META NAME="Author" CONTENT="mon nom">
   <META NAME="Description" CONTENT="ce que c'est">
   <META NAME="Keywords" CONTENT="mots clef associés">
</HEAD>
```

15. Conclusion

Voici ce premier tour d'horizon du HTML terminé. La norme CGI aidant, vous avez tout pour faire des sites WWW Internet graphiques avec toutes les fonctionnalités de l'interaction.

15. Pour aller plus loin ...

McArthur Douglas c, Dr Dobb's Journal #224 dec.1994, World Wide Web and HTML

Davidson Andrew, Coding with HTML forms, Dr Dobb's Journal #231 jun.1995

Davidson Andrew, Clickable image in HTML, Dr Dobb's Journal #234 sep.1995

© Copyright 1998-1999 Emmanuel PIERRE. Libre reproduction sous Licence LLDDv1.

Send all comments to mailto:webmaster@e-nef.com?Subject=Contact&Body=About your Web site e-nef.com

Last Update 05/05/1999