



Hyper Text Markup Language

préambule

- *cette présentation, basée sur la norme HTML3.2, n'a pas vocation à être exhaustive*
- *la notation **X|Y** signifie que l'on peut mettre l'élément **X** ou l'élément **Y** dans l'expression*
- *un étudiant dont le login est **toto** et qui veut rendre accessible une page web à l'url*
<http://www.info.univ-angers.fr/pub/toto/mapageweb.html>
 - *doit créer un répertoire **public_html** dans sa HOME DIRECTORY (ie : **~toto** ∪ **/home/etud/.../toto**)*
 - *doit placer le fichier **mapageweb.html** dans ce répertoire **public_html***
 - *s'assurer que n'importe quel utilisateur*
 - *a le **droit de passage** sur **~toto** et sur **~toto/public_html***
 - *a le **droit de lecture** sur **mapageweb.html***

méthodologie de développement

- Développement sous windows
 - écrire la page web avec le bloc-notes ou le composer de Netscape en l'enregistrant sur le disque D:
 - la visualiser « en local » avec Netscape ou IE par le menu Fichier, ouvrir...
 - la transférer par FTP sur le serveur sirius dans le répertoire public_html
- Développement sous Linux (serveur sirius)
 - Écrire la page web avec Quanta ou asWedit ou n'importe quel éditeur de texte en l'enregistrant dans un répertoire de travail
 - La visualiser en « local » avec Netscape par le menu Fichier, Consulter une page,...
 - La recopier dans le répertoire public_html
- La page est ensuite accessible à l'url
<http://www.info.univ-angers.fr/pub/toto/mapageweb.html>

balises ou marqueurs (tag) HTML

- Une page web est composée d'éléments (textes, images, liens hypertextes, ...) à afficher selon une présentation définie par des balises
- les balises peuvent être écrites en minuscules ou majuscules et vont (presque toutes) par paire :
 - une balise ouvrante `<truc>` et une balise fermante `</truc>`
 - leur action porte sur les éléments placés entre la paire
 - balises simples `<truc>`
 - balises avec attributs `<truc attribut="valeur">....</truc>`
 - les balises peuvent et doivent être imbriquées hiérarchiquement

	<code><truc></code>	<code><truc></code>	
	<code><machin></code>	<code><machin></code>	
écriture	<code></machin></code>	<code></truc></code>	erreur
correcte	<code><truc></code>	<code></machin></code>	

structure générale d'une page web

début

en-tête :
description
du document
et insertion
de scripts

corps :
partie
visible du
document

fin

```
<HTML>
<HEAD>
  <TITLE>ma page web </TITLE>
  <SCRIPT> ... </SCRIPT>
  <META NAME="...." CONTENT=".....">
</HEAD>

<BODY>
  Bla bla bla .....
  <SCRIPT> ... </SCRIPT>
  <!-- commentaires -->
</BODY>
</HTML>
```

mise en page

- `<BODY BGCOLOR="#RVB">....</BODY>` pour un fond d'écran de la couleur R+V+B. R, V et B sont des nombres hexadécimaux compris entre 0 et 255 (FF) représentant la quantité de rouge, de vert et de bleu. `#RVB` peut-être remplacé par le nom standard de la couleur (ex: white)
- `<BODY BACKGROUND="url">....</BODY>` pour un fond d'écran composé de l'image contenue dans le fichier dont l'url est spécifié
- la balise `<BODY>` permet aussi de spécifier des paramètres généraux appliqués sur tout le texte de la page (couleur du texte, couleur des liens, couleurs des liens visités, ...)

mise en page

- `<DIV ALIGN="left" | "right" | "center">....</DIV>` pour créer une division où le texte sera aligné à gauche, à droite ou au milieu
- `<CENTER>....</ CENTER >` version simplifiée pour centrer la présentation
- `<P ALIGN="left" | "right" | "center">....</ P>` pour créer un paragraphe aligné à gauche, à droite ou au milieu
- `<P>` employé seul marque une fin de paragraphe, et provoque donc un retour à la ligne avec saut
- `
` passage à la ligne sans saut

mise en page

- `<HR>` trace une ligne horizontale sur toute la largeur de la page, attributs possibles :
 - `SIZE="N"` pour une épaisseur de N pixels
 - `WIDTH="N"` pour une largeur de N pixels
 - `WIDTH="N%"` pour une largeur de N% par rapport à la fenêtre
 - `ALIGN="left" | "right" | "center"` pour placer la ligne sur la page

mise en forme de texte

- `<BIG>gros</BIG>`
- `<SMALL>petit</SMALL>`
- `gras `
- `<I>italique</I>`
- `^{` mise en exposant `}`
- `_{` mise en indice `}`
- `<PRE>` texte préformaté `</PRE>` le texte est affiché tel que dans le fichier source
- `<H1>` titre de niveau 1 `</H1>` (idem avec `<H2>`, ..., `<H6>` pour des titres de niveaux inférieurs)

choix de la police de caractères

- *Il n'est pas forcément souhaitable de fixer l'utilisation d'une police particulière*
 - *car elle risque de ne pas être disponible sur la plate forme du navigateur*
 - *car l'utilisateur du navigateur a pu choisir sa propre police*
- **** texte ****, attributs possibles :
 - **FACE="Arial,Helvetica"** pour spécifier d'abord Arial, et si non disponible alors Helvetica
 - **SIZE="N" | "+N"** (N compris entre 1 et 7)
 - N valeur absolue
 - +N valeur relative par rapport à la taille standard
 - **COLOR="#RGB"** (idem BODY)

listes

- liste non numérotée

```
<UL>  
<LI> item1 .... </LI>  
...  
</UL>
```

- liste numérotée

```
<OL>  
<LI> item1 </LI>  
...  
</OL>
```

- liste de définitions

```
<DL>  
<DT> terme1 </DT>  
<DD>définition 1</DD>  
...  
</DL>
```

tables

- Elles servent à présenter des tableaux de données mais aussi à faire des effets de présentation

début de table `<TABLE>`

`<TH>` en-tête colonne 1 `</TH>` ...
début de ligne `<TR>`

 ...
début de cellule `<TD>`

 ...
fin de cellule `</TD>`

fin de ligne `</TR>`

 ...
fin de table `</TABLE>`

tables

- attributs pour la balise **<TABLE>**
 - **BORDER="N"** bordure de N pixels d'épaisseur, aucune bordure si N=0
 - **WIDTH="N"** largeur de la table égale à N pixels
 - **WIDTH="N%"** largeur de la table égale à n% de celle de la fenêtre
 - **BGCOLOR="couleur"** pour fixer la couleur du fond de la table
 - **CELLSPACING="N"** espacement entre les cellules
 - **CELLPADDING="N"** espacement entre le contenu d'une cellule et le bord de la cellule
- **<CAPTION >** légende de la table **</CAPTION >**
 - attribut **ALIGN="TOP" | "BOTTOM"** pour placer la légende au-dessus ou en dessous de la table

tables

- Attributs communs à `<TR>`, `<TD>` et `<TH>`
 - `COLOR="couleur"` pour fixer la couleur du fond
 - `ALIGN="left" | "right" | "center"` pour l'alignement horizontal
 - `VALIGN="bottom" | "middle" | "top"` pour l'alignement vertical
- Attributs de `<TD>`
 - `COLSPAN="N"` permet à la cellule de recouvrir N colonnes
 - `ROWSPAN="N"` permet à la cellule de recouvrir N lignes

les url

- Uniform Resource Locator : adresse d'un document (au sens large) sur Internet

`http://www.info.univ-angers.fr/pub/pn/index.html#bas`
=
`protocole://serveur/.../répertoire/.../fichier#ancre`

insertion d'images

- `` insère l'image contenue dans le fichier (format jpeg, gif ou png) dont l'url est spécifié, attributs :
 - `ALT="texte"` texte à afficher si le navigateur n'affiche pas les images, également affiché lorsque la souris passe au-dessus de l'image
 - `WIDTH="N"` largeur de l'image en pixels
 - `HEIGHT="N"` hauteur de l'image en pixels
 - `ALIGN="left" | "right" | "top" | "middle" | "bottom"` pour aligner l'image par rapport au texte
 - `HSPACE="N"` N pixels d'espacement horizontal autour de l'image
 - `VSPACE="N"` N pixels d'espacement vertical autour de l'image

liens hypertextes

- ` texte ` crée un lien qui renvoie vers l'url spécifié lorsque l'utilisateur clique sur le texte
- ` <A>` crée un lien qui renvoie vers l'url1 lorsque l'utilisateur clique sur l'image spécifiée dans url2
- soit une page web d'url U contenant
` X `
qui définit une ancre **toto** positionnée sur X, alors le lien
` Y ` renvoie vers l'endroit où est placé X dans la page web U lorsque l'on clique sur Y (utile pour la navigation à l'intérieur d'une page)
- *quand c'est possible utiliser des url relatives à la page web en cours de construction (ex : ../../rep1/truc.html) plutôt que des url absolues*

formulaire

- un formulaire inclus dans une page web permet à l'utilisateur de saisir des données, faire des choix, ...
 - pour envoyer au serveur web les données saisies par l'utilisateur
 - pour passer en arguments à une fonction Javascript les données saisies par l'utilisateur
- ex : <http://www.info.univ-angers.fr/pub/pn/Html/formulaire.html>
- une même page peut contenir plusieurs formulaires mais on ne peut pas imbriquer un formulaire dans un autre

formulaires

- `<FORM>` éléments du formulaire `</FORM>` si l'on n'a pas besoin d'envoyer de paramètres au serveur web
- `<FORM ACTION="U" METHOD="POST" | "GET">` éléments du formulaire `<FORM>` si l'on veut envoyer les paramètres au "programme" désigné par l'url U.
 - méthode GET : les paramètres sont envoyés concaténés à la fin de l'url U
 - méthode POST: les paramètres sont envoyés dans un flux indépendant entre le navigateur et le serveur web
 - l'action sera déclenchée par le clic sur un bouton de type submit
- on peut nommer le formulaire à l'aide de l'attribut `NAME="toto"`

méthodes GET et POST



éléments de formulaire

- Tous les éléments peuvent contenir un attribut
 - `NAME="toto"` s'il y a besoin de les nommer
 - `VALUE="titi"` s'il y a besoin de lui donner une valeur
- `<TEXTAREA ROWS="R" COLS="C">`
valeur par défaut
`</TEXTAREA>` affiche une zone de saisie de texte de R lignes et C caractères de large
- `<INPUT TYPE="text">` affiche une boîte de saisie de texte
 - `SIZE="N"` de N caractères de large
 - `VALUE="toto"` avec "toto" comme valeur par défaut
 - `MAXLENGTH="N"` saisie limitée à N caractères
- `<INPUT TYPE="password">` affiche une boîte de saisie de texte à l'aveugle (les caractères entrés sont affichés sous forme d'*), mêmes attributs que "text"

éléments de formulaire

- `<INPUT TYPE="hidden">` crée une "zone mémoire" invisible dans la page web
- `<INPUT TYPE="reset">` affiche un bouton permettant de réinitialiser le formulaire lorsque l'on clique dessus (si on lui donne une valeur celle-ci sera affichée)
- `<INPUT TYPE="checkbox">` affiche une case d'une boîte à cocher
 - `CHECKED` pour forcer (indiquer que) la boîte à être (été) cochée
- `<INPUT TYPE="radio">` affiche un bouton radio, lorsque plusieurs boutons de même nom existent un seul peut être coché (ou exclusif entre tous les boutons)
 - `CHECKED` pour forcer (indiquer que) le bouton à être (été) coché

éléments de formulaire

- `<INPUT TYPE="button" value="toto" ONCLICK="action">` affiche un bouton portant le texte "toto" et lançant action lorsque l'on clique dessus
- `<INPUT TYPE="submit" value="toto">` affiche un bouton portant le texte "toto" et soumettant le formulaire lorsque l'on clique dessus (active l'url spécifiée dans l'attribut **ACTION** du formulaire)

éléments de formulaire

- `<SELECT>`
`<OPTION VALUE="toto1" > texte1`
...
`</SELECT>`

affiche un menu déroulant permettant de sélectionner l'une des options, chaque ligne du menu a une valeur spécifique (ex: toto1) et affiche le texte spécifié à côté (ex: texte1)

- `<OPTION VALUE="toto1" SELECTED> texte1`
permet de choisir cette ligne comme option par défaut

- `<SELECT SIZE="N" MULTIPLE>... </SELECT>` affiche une fenêtre de N lignes et permet la sélection simultanée de plusieurs lignes