

Introduction au < HTML >

vers le XHTML et CSS ...

Fuchs P. Maupetit J.

Université Paris VII

25/09/2006

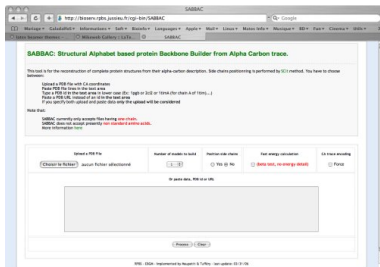
- 1 Objectifs
- 2 Introduction
- 3 Les bases du HTML 4.0
- 4 Les outils HTML avancés
- 5 Vers la norme XHTML/CSS

- 1 Objectifs
- 2 Introduction
- 3 Les bases du HTML 4.0
- 4 Les outils HTML avancés
- 5 Vers la norme XHTML/CSS

Pourquoi ce cours ?

A la fin de ce cours, vous devez posséder les bases pour

- créer votre page perso
- créer votre premier formulaire appelant un CGI



- 1 Objectifs
- 2 Introduction
 - Définition
 - Origines
 - C'est quoi le oueb ?
- 3 Les bases du HTML 4.0
- 4 Les outils HTML avancés
- 5 Vers la norme XHTML/CSS

Definition

HTML : *Hyper Text Mark up Language* ou "Langage de balisage d'hypertexte".

- Langage balise
- Met en forme du texte comme \LaTeX
- Permet de partager des information partout dans le monde
- L'ordre de consultation des données est laissé à l'utilisateur
- **Ce n'est pas un langage de programmation !**

Definition

SGML *Standard Generalized Markup Language*. C'est une norme pour l'échange des documents électroniques. Il a donné naissance au HTML / XML.

HTML

Actuellement en version 4.0, c'est un sous ensemble de SGML.

XHTML

HTML "propre" selon la norme du W3C
<http://www.w3c.org>

XML

Extensible Markup Language

- meta-langage à balises
- version simplifiée de SGML
- permet de définir ses propres balises

- Un **serveur web** est une machine quelque part sur le réseau qui nous permet de mettre à disposition des pages web.
- Un **navigateur web** (Firefox, Opera, Safari et ... IE?)
- On accède à une page web sur un serveur par une **URL** (*Universal Resource Locator*) ou adresse :
protocole `://adressesmachine/rep/sous-rep/document`
- Le **protocole** peut être :
 - `http` le plus courant permet de transfert de pages web (port 80) Ex :
`http://www.google.fr`
 - `ftp` permet le transfert de fichiers (port 21) Ex :
`ftp://debian.ens-cachan.fr/debian-cd/`
 - ...
- **DNS** (*Domain Name Server*) : chaque machine a une adresse IP (*Internet Protocol*) sur le réseau, et les DNS permettent de faire correspondre des noms de domaine (`python.org` par ex) à des adresses IP (en v4 : `134.157.81.34`)

1 Objectifs

2 Introduction

3 Les bases du HTML 4.0

- Généralités
- Structure d'une page HTML
- Mise en forme
- Les listes
- Les "liens hypertexte"
- Insérer une image
- Les caractères spéciaux
- Générer une page HTML

4 Les outils HTML avancés

Site d'intérêt

Une référence en HTML 4.0 : le site de Dave Ragett

<http://www.w3.org/MarkUp/Guide/Overview.html>

Les balises

- Ouverture `<balise>`
- Fermeture `</balise>`
- Encadre du texte `<balise>Texte</balise>`
- Le couple de balises prend effet au niveau du **Texte**
- Insensible à la casse : `<balise>` = `<BALISE>` = `<BaLiSe>`
- Certaines balises n'ont pas besoin d'être fermées.

Structure d'une page HTML

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

- `<HTML>...</HTML>` encadre le document
- `<HEAD>...</HEAD>` en-tête de la page
- `<TITLE>...</TITLE>` donne un titre à la page (moteurs de recherches)
- `<BODY>...</BODY>` corps de la page

Remarques

- Le fichier HTML est un simple fichier texte interprété par votre navigateur.
- La mise en forme est insensible aux retours chariots, sauf dans la balise `<PRE>`

Texte

- souligné : `<U>...</U>`
- gras : `...`
- italique : `<I>...</I>`
- exposant : `^{...}`
- indice : `_{...}`

Titres

- `<H1>...</H1>` du plus gros
- `<H2>...</H2>`
- `<H3>...</H3>`
- ...
- `<H6>...</H6>` au plus petit

Séparateurs

- `<P>` nouveau paragraphe
- `
` retour à la ligne
- `<HR>` ligne horizontale
- **Il n'existe pas de balises fermantes pour les séparateurs.**

La balise <PRE>

Idéale pour les fichiers formatés en colonne ou le code

Code :

```
<pre>
ATOM      5  CA  ARG      1      -0.977  -9.308  24.396
ATOM     22  CA  GLY      2      -1.731  -6.186  23.074
ATOM     27  CA  ASN      3         0.654  -5.826  20.081
ATOM     38  CA  VAL      4      -0.293  -8.716  17.970
</pre>
```

Rendu :

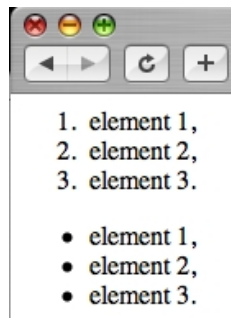
```
ATOM      5  CA  ARG      1      -0.977  -9.308  24.396
ATOM     22  CA  GLY      2      -1.731  -6.186  23.074
ATOM     27  CA  ASN      3         0.654  -5.826  20.081
ATOM     38  CA  VAL      4      -0.293  -8.716  17.970
```

Listes numérotées

```
<OL>  
<LI>element 1,  
<LI>element 2,  
<LI>element 3.  
</OL>
```

Listes à puces

```
<UL>  
<LI>element 1,  
<LI>element 2,  
<LI>element 3.  
</UL>
```



Vous pouvez faire des listes imbriquées.

Les "liens hypertexte"

Les liens externes

` Python `

Ex : `python`

Les liens internes

- Définis au sein d'une même page
- Définition de l'accroche : `...`
- Lien vers l'accroche : `...`

URL absolues / relatives

- Absolues : en général vers un autre site
- Relative : à utiliser au sein du même site (utile lorsqu'on déplace plusieurs pages relatives les unes aux autres)

Site FTP

```
<A HREF="ftp ://ftp.rcsb.org/pub">...</A>
```

Mail

```
<A HREF="mailto :monmail@toto.fr">...</A>
```


La balise image

- `` dans le champ SRC, on peut mettre une image locale ou une URL.
- On peut mettre un lien sur une image
` ...`
- On peut jouer sur les attributs de la balise image, par exemple pour la centrer, on peut faire :
``
- Notez bien que ce genre d'attribut s'applique aussi aux balises paragraphe, liste ...

La syntaxe générale `&code ;` ;

&	<code>&amp;</code>	é	<code>&eacute;</code>	è	<code>&egrave;</code>
à	<code>&agrave;</code>	ê	<code>&ecirc;</code>	espace	<code>&nbsp;</code>

Pour une liste complète, allez sur

<http://www.commentcamarche.net/html/htmlcarac.php3>.

Générer du HTML ?

WYSIWIG : *What You See Is What You Get*

- Dreamweaver, Frontpage ...
- Avantage : visualisation directe
- Inconvéniant : qualité et maintenance du code

Editeurs de texte

Avec la coloration syntaxique et la génération automatique de code, Emacs, VI, NEdit ... sont des outils performants pour avoir un code propre.

Indentation

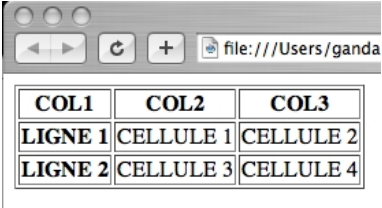
Pensez à indenter votre code pour le rendre plus lisible. Une règle : nouvelle division, nouvelle indentation.

- 1 Objectifs
- 2 Introduction
- 3 Les bases du HTML 4.0
- 4 Les outils HTML avancés**
 - Les tableaux
 - Les cadres
 - Les formulaires
- 5 Vers la norme XHTML/CSS

- Moyen élégant de présenter des données
- balise : `<table>...</table>` définit les limites du tableau qui peut contenir les attributs :
 - `border` : largeur en pixel du cadre autour du tableau
 - `width` : largeur en pixel du tableau
 - `align` : pour placer le tableau (center, left, right)
 - etc
- balise `<TR>` : définit une nouvelle ligne du tableau
- balise `<TH>` : définit les en-têtes du tableau
- balise `<TD>` : définit les données du tableau

Un exemple de tableau

```
<TABLE BORDER=1>  
  <TR>  
    <TH>COL1  
    <TH>COL2  
    <TH>COL3  
  <TR>  
    <TH>LIGNE 1  
    <TD>CELLULE 1  
    <TD>CELLULE 2  
  <TR>  
    <TH>LIGNE 2  
    <TD>CELLULE 3  
    <TD>CELLULE 4  
</TABLE>
```



A screenshot of a web browser window. The address bar shows the file path: file:///Users/ganda. The browser displays a table with a border. The table has three columns and three rows. The first row contains three header cells: COL1, COL2, and COL3. The second row contains three data cells: LIGNE 1, CELLULE 1, and CELLULE 2. The third row contains three data cells: LIGNE 2, CELLULE 3, and CELLULE 4.

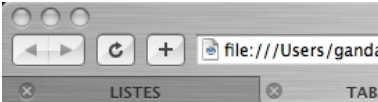
COL1	COL2	COL3
LIGNE 1	CELLULE 1	CELLULE 2
LIGNE 2	CELLULE 3	CELLULE 4

Tableaux : Fusion de lignes

```
<TABLE BORDER=1>
  <TR>
    <TH>COL1
    <TH>COL2
    <TH>COL3
  <TR>
    <TH>LIGNE 1
    <TD>CELLULE 1
    <TD ROWSPAN=2>CELLULE 2
  <TR>
    <TH>LIGNE 2
    <TD>CELLULE 3
</TABLE>
```

ROWSPAN

L'attribut ROWSPAN=N permet la fusion de cellules sur N lignes.



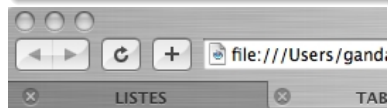
The screenshot shows a browser window with a table. The table has three columns: COL1, COL2, and COL3. The first row contains LIGNE 1, CELLULE 1, and CELLULE 2. The second row contains LIGNE 2, CELLULE 3, and CELLULE 2. The cell 'CELLULE 2' in the first row is visually merged with the cell 'CELLULE 2' in the second row, demonstrating the effect of the ROWSPAN=2 attribute.

COL1	COL2	COL3
LIGNE 1	CELLULE 1	CELLULE 2
LIGNE 2	CELLULE 3	CELLULE 2

```
<TABLE BORDER=1>
  <TR>
    <TH>COL1
    <TH>COL2
    <TH>COL3
  <TR>
    <TH>LIGNE 1
    <TD>CELLULE 1
    <TD COLSPAN=2>CELLULE 2
  <TR>
    <TH>LIGNE 2
    <TD>CELLULE 3
  </TABLE>
```

COLSPAN

L'attribut COLSPAN=N permet la fusion de cellules sur N colonnes.



COL1	COL2	COL3
LIGNE 1	CELLULE 1	
LIGNE 2	CELLULE 3	CELLULE 4

- On peut mettre autre chose que du texte dans un tableau, comme
 - des images (Galerie)
 - des liens
 - des formulaires (non conseillé par le W3C)
- Les balises `<TR>`, `<TH>` et `<TD>` admettent un certain nombre d'attributs parmi lesquels l'attribut `align` (`center`, `left`, `right`).
- Si votre code est correctement indenté, vous allez "lire" facilement les tableaux à partir du source HTML.
- Les éditeurs de texte peuvent générer des tableaux automatiquement (il existe aussi des outils sur le web).

Definition

Les cadres permettent de découper une page HTML en plusieurs zones.

Principe

- Il faut d'abord définir le schéma de la page, *ie* le découpage en zones, grâce à la balise `<FRAMESET>...<FRAMESET>`. Les attributs de la balise `<FRAMESET>` sont :
 - `rows="20%,80%"` : découpage horizontal en 2 zones, exprimé en pourcentage de la taille de la fenêtre du navigateur.
 - `cols="20%,80%"` : découpage vertical en 2 zones, exprimé en pourcentage de la taille de la fenêtre du navigateur.
- La balise `<FRAME>` définit la page HTML qui sera affichée dans le cadre. Ses attributs sont :
 - `name` le nom de la frame (utile pour les liens, javascript)
 - `scrolling=(yes ou no)` barre de défilement ?
 - `src=fichier.html` : le source du cadre

Un exemple de cadre

MaPage.html

```
<HTML>
  <FRAMESET COLS="20%,80%">
    <FRAME NAME="CadreGauche"
      SRC="menu.html">
    <FRAME NAME="CadreDroit"
      SRC="content.html">
  </FRAMESET>
</HTML>
```

menu.html

```
<HTML>
  <BODY>
    MENU
    <BR>
    <H1>CADRE GAUCHE</H1>
  </BODY>
</HTML>
```

content.html

```
<HTML>
  <BODY>
    Contenu de la page.
    <BR>
    <H1>CADRE DROIT</H1>
  </BODY>
</HTML>
```



Les liens actifs dans un autre cadre, l'attribut TARGET

MaPage.html

```
<HTML>
<FRAMESET COLS="20%,80%">
  <FRAME NAME="CadreGauche"
    SRC="menu.html">
  <FRAME NAME="CadreDroit"
    SRC="content.html">
</FRAMESET>
</HTML>
```

menu.html

```
<HTML>
<BODY>
  MENU
  <BR>
  <H1>
    <A HREF="http://www.python.org"
      TARGET="CadreDroit">PYTHON</A>
  </H1>
</BODY>
</HTML>
```

content.html

```
<HTML>
<BODY>
  Contenu de la page.
  <BR>
  <H1>CADRE DROIT</H1>
</BODY>
</HTML>
```



Remarque

Vous ne devez pas définir de balises `<BODY>...</BODY>` dans la page principale d'écrivant les cadres (`<FRAMESET>`) !

- Les cadres sont très pratiques (et étaient souvent employés) pour définir une zone de menu.
- Tous les navigateurs ne les supportent pas.
- Sont de moins en moins utilisés, parce que leur gestion est lourde. Ils vont disparaître.

Definition

Les formulaires permettent d'envoyer des informations au serveur web qui héberge la page qui le contient. En général, ces informations sont ensuite traitées par un script CGI (python, perl, C ...) ou du PHP, ASP ... qui renvoie un résultat au client ayant soumis le formulaire.

La balise <FORM> . . . </FORM>

- Elle encadre le formulaire
- l'attribut METHOD définit le moyen d'envoyer ces informations
 - GET par l'URL
 - POST par l'entrée standard
- L'attribut ACTION pointe sur le script du serveur qui reçoit les informations.

Champ texte

```
<INPUT TYPE="text" SIZE=20 MAXLENGTH=30 NAME="...">
```

Case à cocher

La valeur est mise à ON ou OFF (1 ou 0).

```
<INPUT TYPE="checkbox" NAME="..." VALUE="...">
```

Boutons radio

Ces boutons sont exclusifs : une seule variable correspondant à une seule valeur (si les champs NAME sont identiques). Utile quand peu de valeurs sont possibles.

```
<INPUT TYPE="radio" NAME="..." VALUE="a">
```

```
<INPUT TYPE="radio" NAME="..." VALUE="b">
```

Champ de saisie

Permet de saisir de longs textes avec retour chariot, etc ... ROWS définit le nombre de lignes de ce champ texte, et COLS le nombre de colonnes.

```
<TEXTAREA NAME="..." ROWS=4 COLS=20>
```

```
</TEXTAREA>
```

Liste déroulante

Comme les bouton radios, il n'existe qu'une seule variable donc une seule valeur. Utile quand beaucoup de valeurs sont possibles.

```
<SELECT NAME="...">  
  <OPTION VALUE="a"> texte de la liste avec la valeur 'A'  
  <OPTION VALUE="b"> texte de la liste avec la valeur 'B'  
  <OPTION VALUE="c"> texte de la liste avec la valeur 'C'  
</SELECT>
```

Informations cachées

Ce champ n'est pas visible par l'utilisateur dans le formulaire. Il permet d'ajouter des couples variable/valeur au formulaire. Utile dans le suivi d'une valeur par exemple.

```
<INPUT TYPE="hidden" NAME="..." VALUE="...">
```

Bouton : soumission

```
<INPUT TYPE="submit" NAME="..." VALUE="Texte Du Bouton">
```

Bouton : réinitialiser

```
<INPUT TYPE="reset" NAME="..." VALUE="Texte Du Bouton">
```


- Le champ `NAME` sera reconnu en tant que tel par le script cible sur le serveur : `NAME` sera le nom de la variable, et `VALUE` sa valeur.

Astuce

En HTML, on peut passer des tableaux de valeurs (dans le sens développement du terme). Pour ce faire, votre nom de variable doit être du type `tab[]`, et le navigateur se charge du reste.

- 1 Objectifs
- 2 Introduction
- 3 Les bases du HTML 4.0
- 4 Les outils HTML avancés
- 5 Vers la norme XHTML/CSS
 - Principe
 - Le XHTML
 - Le CSS
 - Validation

On m'aurait menti ?

XHTML : Mettez moi un peu d'ordre dans tout ca !

- Le XHTML a été créé pour faire face à l'anarchie totale qu'a entraîné le HTML
- Il impose un certain nombre de règles strictes issues du monde XML.
- Les attributs sont réduits à leur stricte minimum.
- On sépare la mise en forme de la structure logique du document.

XHTML/CSS : un couple détonnant

- Le XHTML définit la structure logique du document : sections, sous-sections, zones ...
- Le CSS (*Cascade Style Sheet*) définit la mise en forme de la page web. On applique ainsi une *classe* ou un *id* a une zone définie par un couple de balises.

EX : `<p class="commentaire">Ici je tape un commentaire ... </p>`

- Les balises sont écrites en **minuscules** <BODY> != <body>
- Les attributs sont aussi en minuscule, et les valeurs des attributs bordés par des doubles quotes :
- **Toutes les balises doivent être fermées** quand elles bordent du texte. Sinon, elles terminent par />. Ex :

```
<option value="a"></option>  
<td></td> ou <tr></tr>  
<p></p>  
<img src="" ... />  
<input type=... />
```

- Un certain nombre de balises obsolètes ont disparu.
- La balise ... est particulièrement utile pour définir une zone très locale quand on n'a pas de balise la bordant.

Une référence : <http://www.w3schools.com/>

Un exemple XHTML/CSS : XHTML (1)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
<title>:: Maupetit Julien Home Page ::</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<meta name="author" content="Maupetit Julien" />
<link rel="stylesheet" href="./css/gandalfos.css" type="text/css" title="galadalfos" />
</head>
<body>
  <!-- Start main -->
  <div id="main">
    <!-- Start header -->
    <div id="header">
      </div>
    <!-- End header -->
    <!-- Start menu -->
    <div id="menu">
      <table class="navig">
        <tr>
          <td><a href="index.html">Home</a></td>
          <td><a href="research.html">Research</a></td>
          <td><a href="teaching.html">Teaching</a></td>
          <td><a href="links.html">Links</a></td>
          <td><a href="http://www.ebgm.jussieu.fr">Staff</a></td>
        </tr>
      </table>
    </div>
    <!-- End menu -->
```

Un exemple XHTML/CSS : XHTML (2)

```
<!-- Start content -->
<div id="content">
...
</div>
<!-- End content -->

<!-- Start footer -->
<div id="footer">
</div>
<!-- End footer -->

<!-- Start bottom -->
<div id="bottom">
  Last-Update: 30/09/2005
</div>
<!-- End bottom -->

</div>
<!-- End Main -->

</body>
</html>
```



Un exemple XHTML/CSS : CSS

```
* {
  font-family: Verdana, Geneva,
  Arial, Helvetica, sans-serif;
}

body {
  font-size: 8pt;
  color:#ffffcc;
  background-color:#336699;
}

img {border:0;}

a {
  color: #336699;
  text-decoration:none;
}

a:hover{color: #003366;}

#main {
  padding:0.2em;
  font-size:10pt;
}

#header {
  width:800px;
  height:113px;
  margin-left:auto;
  margin-right:auto;
  background-image:
    url('../images/1.jpg');
  background-repeat:no-repeat;
}

#menu{
  width:800px;
  margin-left:auto;
  margin-right:auto;
  background-color:#ffffff;
  font-size:8pt;
  font-weight:bold;
}

#menu .navig{
  border-top:1px solid #cccccc;
  border-bottom:1px solid #cccccc;
  width:100%;
  text-align:center;
}

#content{
  width:800px;
  margin-left:auto;
  margin-right:auto;
  color:#000000;
  background-color:#ffffff;
  background-image:
    url('../images/2.jpg');
  background-repeat:no-repeat;
  background-position:top center;
}

#footer {
  width:800px;
  height:30px;
  margin-left:auto;
  margin-right:auto;
  background-image:
    url('../images/2.jpg');
  background-repeat:no-repeat;
}

#bottom {
  width:800px;
  margin-left:auto;
  margin-right:auto;
  padding-top:0.5em;
  text-align:center;
  font-size:6pt;
}
```

- Faites valider vos pages XHTML par le w3c <http://validator.w3.org/>
- Faites valider vos CSS par le w3c <http://jigsaw.w3.org/css-validator/>
- C'est le meilleur moyen d'apprendre ce qu'il faut faire de ce qu'il ne faut pas faire.

Ultime remarque

Ce n'est pas parce que votre page est valide qu'elle va s'afficher partout de la même manière (ni correctement) !

- Seuls les navigateurs évolués (Firefox, Opera, Safari ...) supportent la norme CSS 2.0. Je vous invite donc à les utiliser et à en faire la promotion pour que les gens arrêtent d'utiliser IE6 qui ne respecte pas les standards du W3C !