XHTML / CSS

Un duo gagnant pour un Web moderne

Jeudi 13 Janvier – Jean-Marie Favreau – Thomas Petazzoni

XML : eXtensible Markup Language

- **Meta-language** : permet de définir des languages
- Normalisé par le W3C (~ 1999)
- Structuration en **balises** avec des **attributs**, imbriquées pour former un arbre

```
<personne id="129">
  <nom>Stallman</nom>
  <prenom>Richard</prenom>
</personne>
```

XML : terminologie

Nom de l'attribut

Valeur de l'attribut

```
<groupe nom="staff">
  <personne id="129">
    <nom>Stallman</nom>
    <prenom>Richard</prenom>
  </personne>

  <personne id="42">
    <nom>Adams</nom>
    <prenom>Douglas</prenom>
    <date naissance="11-03-1952"/>
  </personne>
</groupe>
```

Nom de la balise

Balise sans contenu

A quoi ça sert ?

A tout !

- Protocoles de communication
  - Jabber
  - SOAP
  - XML-RPC
- Formats de fichiers
  - OpenOffice
  - SVG
  - DocBook
  - **XHTML**
- Descriptions
  - Interfaces graphiques avec Glade
  - XMI, description des graphes

Jabber : extrait

```
<message to="clientjabber2@example.com" >
  <subject>test 1449</subject>
  <body>test 1449</body>
</message>
<presence type="unavailable" >
  <status>Logged out</status>
</presence>
```

OpenOffice : extrait

```
<text:p text:style-name="P1">Italique</text:p>
<text:p text:style-name="P2">Gras</text:p>
<text:unordered-list text:style-name="L1">
  <text:list-item>
   <text:p text:style-name="P3">Puce</text:p>
  </text:list-item>
</text:unordered-list>
```

Intérêt du XML

- Représentation de n'importe quel type de données
- Nombreux outils génériques à disposition
  - Éditeurs XML
  - Bibliothèques de lecture (*parsing*)
- Sans ambiguités
  - Facile à analyser
  - Possibilité d'appliquer des transformations (XSLT)
- Validation par rapport à des descriptions de language
  - DTD : Document Type Definition
  - XML Schema

Un langage XML destiné à la description du **contenu** de pages Web

```
<h1>Le Logiciel Libre</h1>

<p>Le <strong>Logiciel Libre</strong> offre
quatre libertés :</p>

<ul>
 <li>Liberté d'utiliser ;</li>
 <li>Liberté d'étudier et de modifier ;</li>
 <li>Liberté de copier ;</li>
 <li>Liberté de redistribuer.</li>
</ul>
```

XHTML : Structure générale

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xml:lang="fr">
  <head>
    <title>Votre titre</title>
    <meta http-equiv="Content-Type"
          content="text/HTML; charset=iso-8859-1" />
  </head>
  <body>
    …votre code…
  </body>
</html>
```

XHTML : balises

** ** : Paragraphe

**** : Emphase (italique)

**** : Forte emphase (gras)

<h1></h1>...<h6></h6> : Titres

<img /> : Image

<a> : Lien

<code></code> : Code

**
** : Retour à la ligne

<hr /> : Ligne horizontale

Tableaux

- **<table></table>** : tableau
- **<tr></tr>** : ligne d'un tableau
- **<th></th>** : cellule d'en-tête
- **<td></td>** : cellule

Structuration logique

Possibilité de décomposer la page en éléments logiques : **balise **.

Les éléments logiques sont des parties indépendantes les unes des autres dont on veut pouvoir définir la présentation.

Que du contenu pour l'instant ... pas de présentation !

Pour la présentation, la deuxième moitié du duo gagnant passe à l'action :

CSS

Cascading Style Sheets

CSS permet de définir pour chaque élément d'une page ou d'un ensemble de pages la mise à forme à appliquer.

Une même CSS pour plusieurs pages.

Plusieurs CSS pour la même page :

- Différentes présentations
- Différentes sorties : écran, imprimante

CSS : la syntaxe

Dans le fichier .html, au niveau de l'entête <head> :

```
<link href="standard1.css"
      rel="stylesheet" type="text/css"
      title="Standard" />
```

Dans le CSS, des sections du type :

```
h1 {
  color: #333;
  background-color: #ffffff;
}
```

On peut vouloir appliquer des styles différents à des éléments de même type dans un document. Ex: un lien plus important, une cellule spécifique ...

Deux attributs utilisables sur tous les éléments :

- **class** : quand plusieurs éléments de la page sont de même nature sémantique ;
- **id** : quand un style doit être appliqué à un élément particulier.

CSS : présentation

Pour positionner les différents composants d'une page :

- chaque élément de la page est dans un avec un **id**, par exemple *entete, menu, corps, pied* ou dans un **class**, par exemple *billet, news, article*
- la CSS applique un style aux différents en fonction de leur **class** ou de leur **id**

div#menu {} : applique le style aux d'id=”menu”

div.billet {} : applique le style aux de class=”billet”

div#menu h1 {}

Style appliqué aux <h1> de la div de classe *menu.*

Ce style « hérite » du style général <h1>.

Même principe pour **div.billet p {}**

Ne pas faire

- Utiliser les **tableaux** pour faire de la présentation
- Utiliser l'attribut ***style*** pour intégrer des bouts de CSS dans le code XHTML
- Utiliser des éléments **non normalisés**, comme *<blink>*
- Écrire un code **mal formé** : éléments mal imbriqués, balises en majuscules, omission de guillemets pour les attributs, utilisation du & sans &
- Utilisation d'**attributs non-valides** : attributs de présentation notamment

Valider avec le validateur du W3C !

Historique rapide

- HTML 2 et 3 : l'époque de la guerre Internet Explorer / Netscape, incompatibilités, balises spécifiques ...

- HTML 4 : début de normalisation, mais peu de possibilités de séparation du contenu et de la présentation. Laxismes sur la syntaxe.

- XHTML : le HTML puissance XML, sans attributs de présentation, celle-ci étant dévolue au CSS

A ne pas faire (bis)

```
<TABLE bgcolor=#fefefe cellspacing=2
cellpadding=10>
<TR>
<TD><font
COLOR=#ff0000>Plop</font></td>
<TR>
<TD><blink>Piout</blink></TD>
</tr>
</Table>
```

Problèmes

- Navigateurs pas tous conformes. Internet Explorer ne respecte pas tout CSS 1.0, alors qu'on en est à l'élaboration de CSS 3.

- Comment recopier les éléments communs des pages sur toutes les pages d'un site, par exemple l'entête, le menu ou le pied de page.
  - Pré-processeurs statiques : GTML, WML, XSLT
  - Langages dynamiques : PHP, Perl, Python, XSLT