

# Ouaib pour les ZZ

Tout ce  
qu'il  
faudrait  
savoir !!!



## AVERTISSEMENT



### THESE SLIDES ARE RATED ZZ

- Le support original est présenté dans un cours
- Certains transparents comportent **volontairement** des erreurs. Certaines peuvent aussi être involontaires
- Il est fortement recommandé d'essayer les exemples proposés.

- La page de garde est honteusement copiée des ouvrages des éditions First.
- Ce cours est libre et réutilisable pour tout à chacun sous la licence EMAILCOUCOUWARE ;-)



## Plan



- HTML
- CSS
- XML
  - Validation : Schémas
  - Présentation : CSS & XSLT
  - Programmation

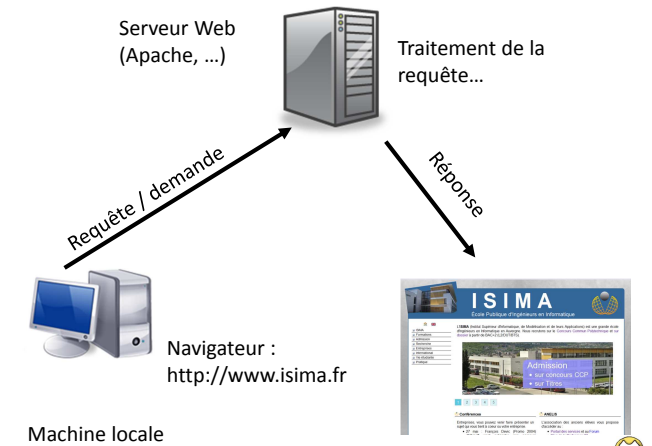


Très courte introduction



## Technologies web

- HTML / ECMAScript
- XML/XHTML
- SGML
- CSS / XSLT
- JAVA
- SCRIPTS CGI, ASP, PHP, JSP...
- Base de données ?



## HTML



## Hyper Text Markup Language

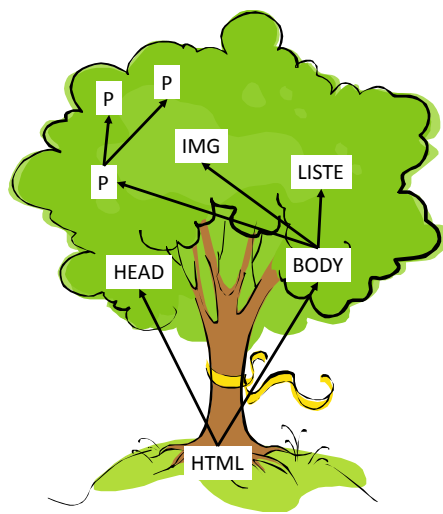
- Langage ?
- Liens hyper texte
- Balises
- Mélange "subtil"
  - Contenu
  - Forme



## HTML

- Compris par un navigateur
  - Tous les navigateurs ne comprennent pas la même chose
- Simple
- Léger en taille
  - PDF ?
  - WORD ?





## Un code tout simple...

```
<html>
  <head>
    <title> Titre </title>
  </head>
  <body>
    un petit contenu
  </body>
</html>
```

Les navigateurs acceptent souvent du code mal écrit !!!  
Mais pas les outils de validation

## Simple mais ordonné...

- Balise <html>
  - Précédé éventuellement par des nouvelles balises (doctype par ex)
- Entête <head>
  - Informations sur la page
  - Titre, jeu de caractères, style
- Corps <body>
  - Contenu de la page exclusivement

## Des balises en ordre !

```
<balise1>
  <balise2>
  </balise2>
</balise1>
```

```
<balise1>
  <balise2>
  </balise1>
</balise2>
```

```
<balise_simple>
```

Liste très limitée de balises, fermantes ou non !  
Insensible à la casse (minuscules conseillées)

## Attribut

- Modification, ajout sur une balise
- Liste restreinte dépendante de la balise
- Insensible à la casse mais minuscules conseillées




```
attribut="valeur"
attribut='valeur'
```

OFFICIEL

```
attribut=valeur
attribut
```

A EVITER

## Éditeurs

- WYSIWYG
  - Adobe Dreamweaver 
  - Front Page > Microsoft Expression 
  - Nvu > KompoZer
  - BlueGriffon 
  - Intégration avec d'autres langages ?

- Un simple éditeur de fichier ?
  - Coloration syntaxique
  - Code source ?

## <p>Paragraphe</p>

```
<p align="left">
  <p align="center">
    <p align="right">
<p align="justify">
```

- Balises de type bloc
- <br> : saut de ligne

## Caractères spéciaux (1)

- Par défaut, les caractères accentués ne sont pas supportés par les navigateurs
- Solutions
  - Utiliser des "entités"
  - Ajouter une balise dans l'entête :

```
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
```

## Caractères spéciaux (2)

&acute;	é	&amp;	&
&Egrave;	È	&quot;	"
&agrave;	à	&apos;	'
&ecirc;	ê	&lt;	<
&iuml;	ï	&gt;	>
		&copy;	©

Liste non exhaustive...



Obsolète

## Police

- `<font> </font>`
- `<font color="blue">`
- `<font size="+2">`
- `<font size="-2">`
- `<font face="Courier">`

Choisir une police par défaut ou une liste de polices :  
essayer de prendre des polices qui existent sur les différents systèmes  
d'exploitation



## Gestion des "blancs"

- 1 ou plusieurs blancs = "1 blanc"
- Saut de ligne pas important
- Espace insécable : `&nbsp;`
- VRAI retour à la ligne : `<br>`
- Paragraphe vide  
`<p> &nbsp; </p>`

Code



## Organisation

- Titres
  - de `<h1>` à `<h6>`
- Listes
  - à puces `<ul>`
  - numérotées `<ol>`
  - élément d'une liste `<li>`
  - Listes imbriquées
- Liste de définition `<dl>`
  - Titre `<dt>`
  - Description `<dd>`



## Titres

- Structuration du document
- Organisation hiérarchique
  - Haut niveau `<h1> ... </h1>`
  - ...
  - Bas niveau `<h6> ... </h6>`
- Balises de type bloc



## Liste non ordonnée

- Changer la puce
  - Attribut `type`
  - `square`, `circle`, `disc`, `none`
  - CSS : toute image
- Balise bloc
- Utilisation CSS
  - menus

```
<ul>
  <li> ... </li>
  <li> ... </li>
  <li> ... </li>
</ul>
```

```
• puce 1
• puce 2
• puce 3
```



## Liste ordonnée

- Changer la numérotation
  - Attribut `type` :  
A, a, I, i, 1
- Début de numérotation
  - Attribut `start`

```
<ol>
  <li> ... </li>
  <li> ... </li>
  <li> ... </li>
</ol>
```

```
1. puce 1
2. puce 2
3. puce 3
```



## Liste de définitions

```
Mot 1
définition 1a
définition 1b
Mot 2
définition 2
```

```
<dl>
  <dt>Mot 1</dt>
  <dd>Def 1a</dd>
  <dd>Def 1b</dd>
  <dt>Mot 2</dt>
  <dd>Def 2</dd>
</dl>
```



Obsolète

## Visuels

- `<u> </u>`
- `<b> </b>`
- `<i> </i>`
- `<blink> </blink>`
- `<marquee> </marquee>` Pas standards !
- `<body bgcolor="navajowhite">`



## Image



- Images JPEG, GIF, PNG
- Balise simple `<img>`
- Attribut `src`
  - `src="en_construction.gif"`
- Attribut `alt`
- Attribut `title`



## Tableau (1)

```
<table>
  <tr>
    <td> L1C1 </td>
    <td> L1C2 </td>
    <td> L1C3 </td>
  </tr>
  <tr>
    <td> L2C1 </td>
    <td> L2C2 </td>
    <td> L2C3 </td>
  </tr>
</table>
```

L1C1	L1C2	L1C3
L2C1	L2C2	L2C3



## Tableau (2)

- height, width : px, em, %
- border
- border="0"
- align : center, right, left



## Balise TD

valign='top' align='left'	valign='top' align='center'	valign='top' align='right'
valign='middle' align='left'	valign='middle' align='center'	valign='middle' align='right'
valign='bottom' align='left'	valign='bottom' align='center'	valign='bottom' align='right'



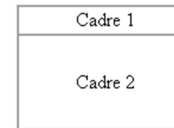
## Les liens ;-)

- <a href="page.html"> mon lien </a>
- target = \_blank, cadre ....
- <a name="ancre"> </a>
- <a href="#ancre"> vers l'ancre </a>
- <a href="page.html#ancre"> retour</a>

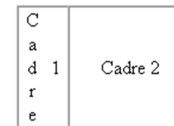


## Cadres / frameset

```
<FRAMESET ROWS="20%,80%">
  <FRAME NAME="Cadre1" SRC="page1.htm">
  <FRAME NAME="Cadre2" SRC="page2.htm">
</FRAMESET>
```



```
<FRAMESET COLS="20%,80%">
  <FRAME NAME="Cadre1" SRC="page1.htm">
  <FRAME NAME="Cadre2" SRC="page2.htm">
</FRAMESET>
```



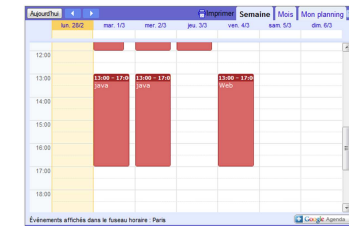
A éviter : préférer iframe

```
<A HREF="page3.htm" TARGET="Cadre1"> Vers la page 3 </A>
```



## iframe

- src, id, name
- width, height, frameborder



## Formulaires

- Utilité avec un langage de script serveur
  - Récupération des informations
  - Vérification et traitement spécifique
- Contrôle navigateur par javascript
- <form> : name, action, method
- <input> : type
- <textarea> : name rows cols
- <select> <option>
- <fieldset> <legend> <label>



## <input>

- Type
  - text, password, hidden
  - radio, checkbox
  - submit, reset, button **checked="checked"**

```
<input type="radio" name="a" value="v1">
<input type="radio" name="a" value="v2">
```

```
<input name="texte" type="text" size="10"
value="essai">
```

```
<input type="submit" >
```

```
<label for="velo">velo</label> id
```



Vous êtes venus en :

- Vélo
- Tram
- Voiture
- À pied
- Aujourd'hui seulement
- Toute la semaine

Envoyer

Même nom  
name = "locomotion"

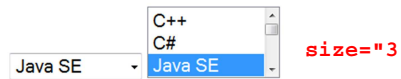
Autre nom  
name = "quand"

```
<form>
<p>Vous êtes venus en :</p>
<input type="radio" name="locomotion" value="velo"> Vélo<br>
<input type="radio" name="locomotion" value="tram"> Tram<br>
<input type="radio" name="locomotion" value="voiture"> Voiture<br>
<input type="radio" name="locomotion" value="pied"> &Agrave; pied
<p>Quand</p>
<input type="radio" name="quand" value="aujourd'hui"> Aujourd'hui<br>
<input type="radio" name="quand" value="toute la semaine"> Toute la semaine
<input type="submit" value="Envoyer">
</form>
```



## <select>

```
<select name="cours"> multiple size
<option>C++</option>
<option>C#</option>
<option>Eiffel</option>
<option selected="selected">Java</option>
<option>Objective C</option>
<option>Smalltalk</option>
</select>
```



## HTML/XHTML

- Balises toujours fermées et en minuscules  
`<br />` `<img />` `<hr />`
- Attributs  
`style="color:black" checked='checked'`
- Préambule de documents
- Meilleure séparation contenu/présentation
  - Balises obsolètes : `<font>`, visuelles
  - Attributs obsolètes
  - CSS



## Doctype

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```



## HTML 5 ?



- En cours de développement. Portabilité ?
- Balises sémantiques
  - article, details, footer, header, mark, section, summary
- Média
  - audio, video, embed
- Autres
  - Canvas (dessin avec ECMAScript)
  - Éléments de formulaire et de contrôle
  - Stockage de données (local ou session)



# ISIMA

## CSS



## Cascade Style Sheets

- Séparer le contenu de la forme
- Abolition des tableaux
  - Sauf données tabulaires ;-)
- Balises spécifiques
  - `<div>` `<span>`
  - Attention au bazar
- Version et compatibilité des navigateurs ?
  - IE 5 – 9, FF, Safari, Chrome, Opera, Konqueror

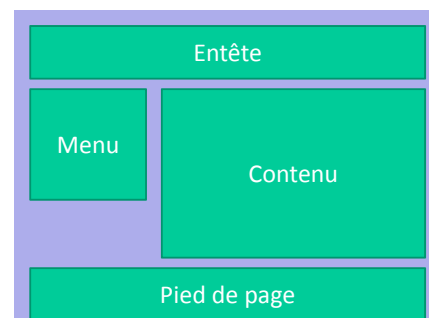


## Visuels II

- Lié à la sémantique
  - `<em>` `</em>`
  - `<strong>` `</strong>`
- Lié à la forme
  - `<span style="color:red">` `</span>`
  - `<p class="justify">` `</p>`
  - `<div id="entete">` `</div>`



## Présentation de site



<http://www.alscreations.com/static/gabarits/index.html>



## Couleur

- Nom
  - 17 + 130
- Rouge Vert Bleu
  - `FF 00 00`
- `rgb(255, 0, 0)`
- `rgb(100%, 0, 0)`
- `#FF0000` ou `#F00`



Tiré de <http://colorshemesigner.com/>



## Différents médias

- Écran : screen
- Imprimante : print
- Présentation : projection
- Braille : braille / embossed
- Oral : aural / speech



## Exemple

```
h1 {
  font-size   : 110%;
  color       : yellow;
  text-align  : center;
  padding-left : 1em;
}

p {
  text-align  : justify;
}
```



## Avoir du style (1)

- Par héritage du contenant
- Par type de balise
- Par classe d'éléments
- Par identification unique
- Par pseudo classe



## Avoir du style (2)

1. Import de fichier(s) CSS
2. Balise <style> dans l'entête
3. Attribut style



```
<a style="font-weight:bold" href="..."> </a>
```



## Fichier texte externe

```
<link rel="stylesheet"
      type="text/css"
      href="fichier.css"
      media="screen" / >
```

```
<style type="text/css">
  @import url(fic1.css);
  @import url(fic2.css) print;
</style>
```

Média "screen" par défaut



## Entête de fichier

```
<style type="text/css">
  <!--
    p {
      text-align : justify;
      font-weight : normal;
    }
  -->
</style>
```



## Classe

Style

```
.center { text-align : center; }
a.petit { font-size : 80%; }
```

Code HTML

```
<p> texte par défaut </p>
<p class="center"> texte centré </p>
```



## Identificateur

Style

```
#entete { text-align : center; }
#baspape { font-size : 80%; }
```

Code HTML

```
<div id="entete"> Ent&ecirc;te </div>
<div id="baspape"> copie </div>
```



## Pseudo classe

- a:hover
- a:visited
- a:first-letter
- a:first-line
- ...



## Fichier texte externe

```
h1, h2, h3 {  
  color : white;  
  background-color : blue;  
}
```

```
a { color : blue; }  
#menu a {  
  color : red;  
}
```



## Taille

- Taille fixe
  - point (pt) pica (pc)
  - centimètre (cm) millimètre (mm)
  - pouce (in)
- Taille relative
  - Cadratin (em) : largeur du "m"
  - hauteur de x (ex)
  - pourcentage (%)
  - pixel (px)



## Police

- font-family
  - 'Arial', 'Trebuchet MS', times, verdana
- font-weight
  - normal bold lighter bolder
  - De 100 à 900
- font-size
  - 14px 200% 3em 0.33em
  - xx-small x-small small medium large x-large xx-large
- font-style
  - normal oblique italic



## Texte

- text-decoration
  - underline overline line-through none
- text-transform
  - capitalize lowercase uppercase none
- text-align
  - left right center justify normal



## Espacements

- Interligne
  - line-height
- Crénage/interlettrage
  - letter-spacing
- Espace entre les mots
  - word-spacing



## Éléments

- En ligne
  - Affichés au fil du texte
  - Pas de positionnement précis prévus
  - Marges inexistantes
- Type bloc
  - Dimensions (hauteur, largeur, profondeur)
  - Sortie du flux normal ?
  - Marges internes (padding) et externes (margin)
  - Bordures
- Changer de type
  - Attribut display : bloc inline none



## Bordures

- border
  - border-top / border-right / border-bottom / border-left
- border-style
  - dashed dotted double groove hidden inset outset ridge solid none
  - Tout le cadre ou bord par bord (T R B L)
- border-width
  - thin medium thick
  - Nombre (1, 2, 3, 4)
- border-color



## Listes

- List-style-type
  - none, disc, circle, square
  - decimal, lower-alpha (a), lower-roman (i), upper-alpha (A), upper-roman (I)
  - armenian, decimal-leading-zero, georgian, lower-greek, lower-latin, upper-latin (IE avec DOCTYPE)
- List-style-image : url(fichier)
  - Comportement différent suivant navigateurs
  - <ul> type : none
  - <li> background-image/repeat/position

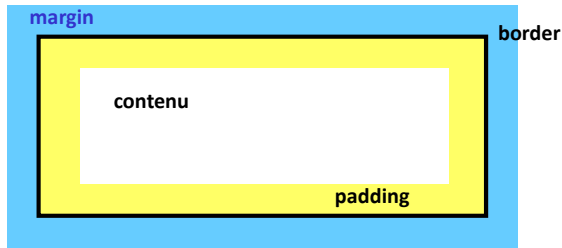


## Positionnement

- Modèle de boîte
- Sortie de flux
  - Absolu
  - Relatif
  - Fixe
  - Flottant
- Profondeur



## Modèle de boîte



- Tous les composants ont des marges par défaut
- *Reset CSS style*



## Différents modèles de boîte

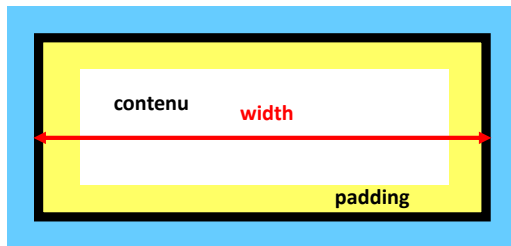
- Modèle standard  
Largeur à l'écran = largeur spécifiée + padding + border
- Modèle Microsoft (<5 et 6 parfois)  
Largeur à l'écran = largeur spécifiée = largeur modulable + padding + border



## Modèle de boîte standard



## Modèle de boîte Microsoft



Modèle par défaut si la page HTML ne contient pas de DOCTYPE  
IE ≥ 6



## Flux normal

- Ordre dans lequel apparaissent les éléments
- Blocs
  - les uns sous les autres
  - utilisent toute la largeur disponible
- Éléments en ligne
  - se suivent pour compléter une ligne

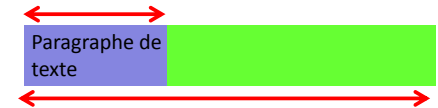


```
<div><p>Paragraphe de texte</p></div>
```

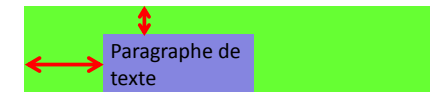
1/ Mettre en bleu p et en vert le div ...

Paragraphe de texte ?

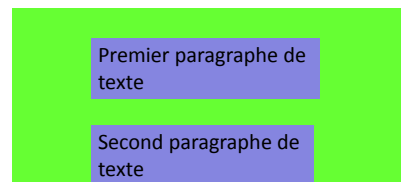
2/ Mettre une largeur de 300px au div et 100px au p



3/ Ajouter un padding-top d'1 em au div et un margin-left de 3em à p



```
<div>
<p>Premier paragraphe de texte</p>
<p>Second paragraphe de texte</p>
</div>
```



- Changer les couleurs de fond et Identifier les paragraphes
- Jouer avec les marges.
- Essayer de les coller en hauteur



## Positionnement relatif

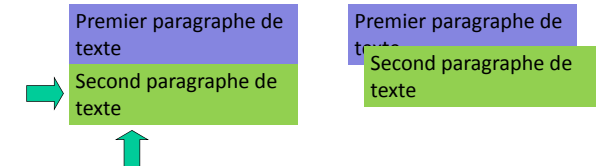
```
position : relative
```

- Élément placé dans le flux normal
- Décalé ensuite grâce à
  - top, right, bottom, left
- Pas d'impact sur le reste du document



```
#second {
...
bottom : 1em;
left: 1em;
}
```

```
<p id="second">
```



## Positionnement absolu

`position : absolute`

- Sortie du flux complète
  - top, right, bottom, left
- Positionnement par rapport
  - Au parent s'il est **positionné**
  - Au premier conteneur **positionné**
  - Body en dernier ressort
  - **Positionné** = relatif, fixe ou absolu



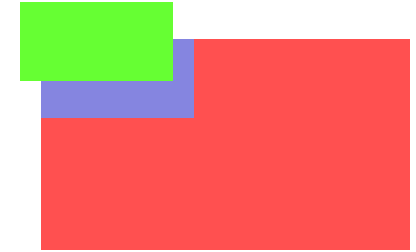
```
<div>
  <p id="prem">Premier paragraphe de texte</p>
  <p id="second">Second paragraphe de texte</p>
</div>
```

```
p#prem {
  width:100px;
  background-color : blue;
}
```

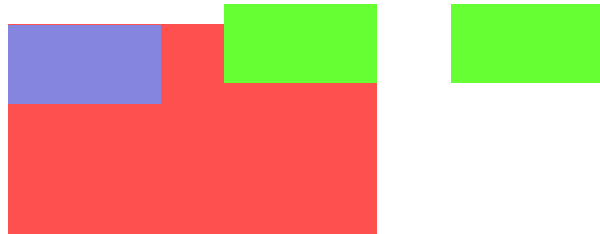
```
p { margin : 0; }
div {
  width : 300px;
  height : 200px;
  background-color : red;
}
```

```
p#second {
  width:150px;
  background-color: green;
}
```

```
p#second {
  position : absolute;
  height : 50px;
  top : 0;
  left:0;
}
```



```
p#second {
  right:0;
}
```



Solution ? Positionner le div englobant (relative)



## Positionnement fixe

`position : fixed`

- L'élément ne défile plus avec la page
- FF ou IE7

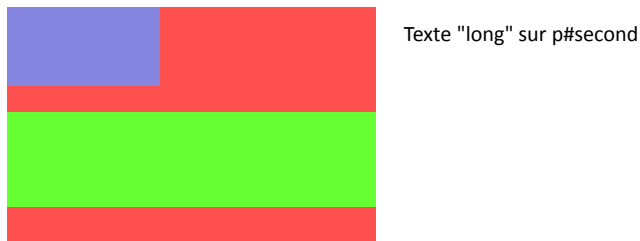


## Positionnement flottant

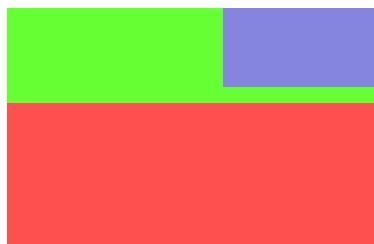
`float : left`

`float : right`

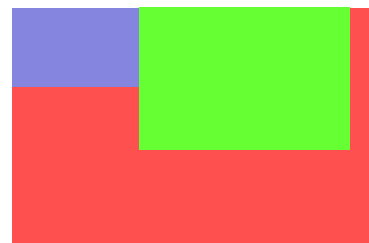
- Bloc retiré du flux pour être placé à gauche ou à droite du bloc qui le contient
- Sa place est alors "libre"
- Attention à la compatibilité !!!
- Empêcher un flottant
  - clear : left right both



float:right sur premier



float : left sur premier et second à largeur fixée



- Que se passe t'il si la hauteur du div est petite (20px) ?
- Comportement non standard sous IE...



## Profondeur

- Comportement par défaut
  - Dernier élément affiché au dessus des autres
- Propriété z-index
  - Seules sont importantes les valeurs relatives

## Reset sheet

- Marges/Ratios différents suivant les navigateurs

```
{
  margin : 0; padding : 0; font-size : 1em;
}
```



## CSS 3

- Toujours en développement
- Compatibilité entre navigateurs ?
  
- Sélecteurs
- Bordures : ombre, cercle
- Transformations 2D et 3D
- Des animations
- Modifications par l'utilisateur



## Outils d'aide

- Extensions
  - Firebug (FF / Chrome)
  - Web Developer (FF / Chrome)
  - MS IE Developer Toolbar
- Scripts de validation W3C
- IE 9 : produit intégré



ISIMA

XML



## eXtensible Markup Language

- Langage générique qui permet d'écrire des langages à balises
  - XHTML
  - MathML
  - SVG
- XML 1.0 en 1998
- But : promouvoir/utiliser le SGML
- 2011 : Support de certaines technologies encore limité



## SGML

- Standard Generalized Markup Language
- Créé en 1986 ;-)
- Séparation forme/contenu
- Trop complexe ?



## Fichier XML

- Fichier texte, plat ?
  - Règles d'écriture précises
  - Document bien formé
- Structure arborescente
  - Système de fichier classique
  - Une racine
- Directives supplémentaires
  - Entête de document
  - Consignes de présentation
- Validation



```
<collection>
<serie nom="Trolls de Troy">
  <bd>
    <titre tome="1">histoires trolls</titre>
    <auteur>Arleston</auteur>
    <dessin>Mourier</dessin>
    <comment>Peur de rien !!!</comment>
  </bd>
  <bd>
    <titre tome="2">le scalp du vénérable</titre>
    <auteur>Arleston</auteur>
    <dessin>Mourier</dessin>
    <comment>Peur de rien !!!</comment>
  </bd>
</serie>
</collection>
```



## Type de nœud (1)

- Texte
  - Non vide
  - Feuille dans l'arbre
- Élément
  - Nom
  - Groupe logique
- Attribut
  - A pour parent un élément
  - Nom + valeur



## Type de nœud (2)

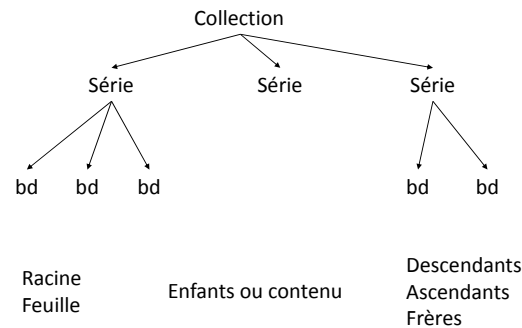
- Commentaire

```
<!-- -->
```

- Instructions de traitement – *processing node*
  - Cible + valeur
- Racine
  - Représente le document entier : commentaires + instructions de traitement
  - Contient l'élément racine



## Arbre XML



## Document bien formé

- Syntaxiquement correct
- Respect de la structure arborescente
 

```
<a><b></a></b>
```
- Respect du nom des balises (casse)
 

```
<collection></Collection>
```
- Respect des conventions d'écriture des attributs
 

```
<serie nom = "Kiwi power">
```
- Utilisation d'entités pour certains caractères



## Clauses supplémentaires

- Entête

```
<?xml version="1.0" standalone="yes" encoding="UTF-8" ?>
```

- UTF-8, UTF-16, ISO-8859-1
- Échapper les caractères : &amp; ;&#0169;

- Style

```
<?xmlstylesheet type="text/xsl" href="fichier.xsl" >
```

- DTD + attribut *standalone*

```
<!DOCTYPE >
```



## Espace de nommage (1)

```
<bd>
<titre tome="1">histoires trolles</titre>
<auteur>Arleston</auteur>
<dessin>Mourier</dessin>
<comment>
  <head>
    <title>Trolles 1</title> XHTML...
  </head>
  <body><p>Peur de rien !!!</p></body>
</comment>
</bd>
```



## Espace de nommage (2)

```
<collection xmlns="http://www.kiux.net">
...
<comment
xmlns:xhtml="http://www.w3.org/TR/xhtml11">
  <xhtml:head>
    <xhtml:title>Trolles 1</xhtml:title>
  </xhtml:head>
  <xhtml:body>
    <xhtml:p>Peur de rien !!!</xhtml:p>
  </xhtml:body>
</comment>
```



## Validation

- Est-ce que le fichier XML suit des règles précises ?
  - Présence / ordre des éléments
  - Attributs
  - Valeurs pour les balises, attributs ?
- DTD
  - *Document Type Definition*
  - Définition de Type de Document
- Schémas XML / *XML Schema Definition (XSD)*
- Autres : Relax NG, ...



## DTD & éléments

```
<!ELEMENT A (B)>
<!ELEMENT A (B, C)>
<!ELEMENT A (C, B)>
<!ELEMENT A (B | C) >
<!ELEMENT A (B, (C | D), E)>
<!ELEMENT A (B | C | (D, E))>
<!ELEMENT A (B?, C)>
<!ELEMENT A (B, (C | D)+ , E) >
<!ELEMENT A (B|C)* >
<!ELEMENT A (#PCDATA, B)* >
```

- Composé mixte
  - #PCDATA apparaît en premier
- LIMITE :**  
Niveau de spécification le plus fin  
*Parsed Character Data*

```
<!ELEMENT A EMPTY >
<!ELEMENT A ANY >
```



## DTD & attributs

```
<!ATTLIST A
  a CDATA #IMPLIED
  b CDATA #IMPLIED >
```

```
<!ATTLIST A a CDATA #IMPLIED >
<!ATTLIST A b CDATA "b" >
<!ATTLIST A c CDATA #REQUIRED >
<!ATTLIST A d CDATA #FIXED "d" >
```

```
CDATA
choix (oui | non)
ID / IDREF
```

```
ENTITY / ENTITIES
NMTOKEN / NMTOKENS
NOTATION
```



```
<collection>
  <serie nom="Trolls de Troy">
    <bd>
      <titre tome="1">histoires trolles</titre>
      <auteur>Arleston</auteur>
      <dessin>Mourier</dessin>
      <comment>Peur de rien !!!</comment>
    </bd>
    <bd>
      <titre tome="2">le scalp du vénérable</titre>
      <auteur>Arleston</auteur>
      <dessin>Mourier</dessin>
    </bd>
  </serie>
</collection>
```

```
<!ELEMENT collection (serie)*>
<!ELEMENT serie (bd)*>
<!ELEMENT bd (titre, auteur,
               dessin, comment?) >
<!ELEMENT auteur (#PCDATA) >
```



## Lier données et DTD

- DTD interne

```
<!DOCTYPE racine [
  <!ELEMENT nom (#PCDATA) >
  <!-- ... -->
]>
```

- DTD externe

```
<!DOCTYPE racine SYSTEM "fichier.dtd" >
```



## XML Schéma

- Alternative XML de la DTD
  - Outils classiques (parseur, XSLT)
  - Non figé
  - Plus puissant
- Norme extensible si besoin
- Contraintes sur une balise ou un attribut
  - Type de données (DATE)
  - Héritage
  - Cardinalités plus fines
  - Facettes, *patterns*, conversion
- Namespaces



```
<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="head" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



## Lier XML et schéma

```
<?xml version="1.0"?>
<note xmlns="http://www.w3schools.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3schools.com note.xsd">
  <to>Tove</to>
  <from>Jani</from>
  <head>Reminder</head>
  <body>Ne pas m'oublier</body>
</note>
```

```
<note xmlns="http://www.w3schools.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file:note.xsd">
```



## Élément simple

- Qui ne contient pas d'autre élément ou d'attribut
- Prédéfini
  - string, decimal, integer, long, boolean, date, time, ...
- Personnel
  - Sous type prédéfini
  - Limitation (facette)
- default/fixed

```
<xs:element name="naissance"
  type="xs:date"/>
```

```
<xs:element name="couleur"
  type="xs:string" default="red"/>
```



## Exemples de Facettes

```
<xs:simpleType name="pourcentage">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="100"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:restriction base="xs:string">
  <xs:enumeration value="velo"/>
  <xs:enumeration value="pied"/>
  <xs:enumeration value="tram"/>
</xs:restriction>
```

```
<xs:restriction base="xs:string">
  <xs:pattern value="[a-zA-Z][a-zA-Z][a-zA-Z]"/>
</xs:restriction>
```



## Élément complexe (1)

- Vide
- Texte seulement
- Autres éléments
- Texte et autres

Avec ou sans attribut(s)

```
<xs:element name="serie">
  <xs:complexType>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="serie"
  type="stype" />
<xs:complexType name="stype">
</xs:complexType>
```



## Élément complexe (2)

- Vide
  - Il suffit de lister les attributs si besoin
- Texte seulement (avec extension ou restriction)

```
<xs:complexType name="stype">
  <xs:simpleContent>
  <xs:restriction>
  </xs:restriction>
  <xs:simpleContent>
  </xs:complexType>
```

- Éléments + texte

```
<xs:complexType name="stype" mixed="true"> ...
</xs:complexType>
```



## Éléments complexes (3)

`<xs:all>`

- L'ordre ne compte pas
- Les éléments apparaissent 0 ou 1 fois

`<xs:choice>`

- Un des éléments

```
<xs:element maxOccurs="1"
  minOccurs="0" />
```

`<xs:sequence>`

- Éléments dans l'ordre

`<xs:any minOccurs="0" />`

- N'importe quel élément

Possibilité de faire des groupes et des groupes d'attributs



## Attribut

- Comme un élément simple
- A placer après les éléments complexes

```
<xs:attribute name="couleur"
  type="xs:string" default="red" />
```

- Optionnel
  - *use* : optional /required (par défaut)
  - *default* : doit respecter le type
  - *fixed*
- Groupe d'attributs et anyAttribute



## XPATH

- Langage pour naviguer dans le document
  - 1.0 et 2.0 (2007)
  - Sous-ensemble commun de XSLT et XQuery
- Fonctions applicables
- Un élément peut être identifié de manière unique
- Forme normale/compacte

```
info0/info1/info2/info3/
axe::nœudtest [exp1] [exp2] ...
```



## Étape de localisation – axe

- child
  - sauf attributs
  - par défaut
- descendant
- parent (unique)
- ancestor
- following-sibling
- preceding-sibling
- following (dans le document)
- preceding (dans le document)
- attribute
- self
- descendant-or-self
- ancestor-or-self



## Exemples

```
n1      tous les enfants de n1
/n0     le nœud racine
/n0/n1  tous les n1 enfants de n0
//n1    tous les n1
n0//n1  tous les n1 enfants de n0
//@lang  tous les attributs lang
```

```
n0/n1 [1]    Premier n1
/n0/n1 [last()]
/n0/n1[@lang]
/n0/n1[@lang="fr"]
/bib/livre[prix>7.0]/titre
../freres
```



## Présentation

- CSS
- XSLT 1.0 & 2.0



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="comics.css"?>
<collection>
  <serie nom="Trolls de Troy">
    <bd>
      <titre tome="1">histoires trolles</titre>
      <auteur>Arleston</auteur>
      <dessin>Mourier</dessin>
      <comment>Peur de rien !!!</comment>
    </bd>
    <bd>
      <titre tome="2">le scalp du vénérable</titre>
      <dessin>Mourier</dessin>
      <auteur>Arleston</auteur>
      <comment>La suite...</comment>
    </bd>
  </serie>
  <serie nom="Largo Winch">
    <bd>
      <titre tome="1">Largo Winch</titre>
      <auteur>Van Hamme</auteur>
      <comment>Inspiré d'une mémoire dans la peau</comment>
    </bd>
  </serie>
</collection>
```



```
serie {
  display : block;
  background-color:#1111FF;
  padding : 0.25em;
  margin:0.5em;
}
bd {
  display : block;
  background-color:#;
  padding : 0.5em;
  margin:1em;
}
auteur, dessin, comm
display:block;
margin-left : 3em;
}
```

```
histoires trolles
  Arleston
  Mourier
  Peur de rien !!!

le scalp du vénérable
  Mourier
  Arleston
  La suite...

Largo Winch
  Van Hamme
  Inspiré d'une mémoire dans la peau
```



## XSLT

- Langage de transformation
  - D'un arbre XML vers un autre
  - Si transfo en HTML, c'est du XHTML ;-)
- 2 versions
  - Version 1.0 supportée par les navigateurs
- Opérations basiques
  - Tri
  - Filtre



## Lien données/fichier XSL

```
<?xml-stylesheet type="text/xsl" href="comics.xsl"?>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<xsl:stylesheet version="2.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  xmlns:nsp="espace de nommage"  
  xmlns="http://www.w3.org/1999/xhtml">
```



## Patrons / templates

- Appliquer un traitement à des nœuds particuliers (XPath)

```
<xsl:template match="/">  
  ...  
  <xsl:value-of select="." />  
  ...  
</xsl:template>
```

```
<xsl:apply-templates select="xpath"/>
```



## Patrons nommés

```
<xsl:template name="filtering">  
  <xsl:param name="filter" select="'none'"/>  
  ...  
</xsl:template>
```

```
<xsl:call-template name="filtering">  
  <xsl:with-param name="filter" select="..."/>  
</xsl:call-template>
```



## Tris

```
<xsl:for-each  
  select="//bd" order-by="+parution">  
  ...  
  // order facultatif, - permet changer de sens  
</xsl:for-each>
```

```
<xsl:apply-templates select="bd">  
  <xsl:sort select="parution"  
    data-type="number" order="descending" />  
  <xsl:sort select="titre" />  
</xsl:apply-templates>
```



## Filtres et conditions (1)

```
<xsl:for-each select="//bd[@tome=1]">  
  ...  
</xsl:for-each>
```

Opérateurs := != &LT& &GT&

```
<xsl:if test=".[@tome='1']">  
  ...  
</xsl:if>
```

```
<xsl:if test="name()='balise'">  
  ...  
</xsl:if>
```



## Filtres et conditions (2)

```
<xsl:choose>  
  <xsl:when match="...">  
    ...  
  </xsl:when>  
  <xsl:when match="...">  
    ...  
  </xsl:when>  
  <xsl:otherwise>  
    ...  
  </xsl:otherwise>  
</xsl:choose>
```



## Variables

```
<xsl:variable name="chemin1"  
  select="document('biblio.xml')/biblio/*"/>
```

```
<xsl:value-of select="count($chemin1)"/>  
<xsl:apply-templates select="$chemin1"/>  
Écrire parfois {$chemin1}
```



## Modification de l'arbre XML

```
<xsl:element>
```

```
<xsl:attribute>
```

```
<xsl:comment>
```

```
<xsl:processing-instruction>
```

```
<xsl:text>
```

```
<xsl:copy> <xsl:copy-of>
```



**ISIMA**

XML & programmation (Java)  
DOM, SAX, ...



## XML et programmation

- Simple fichier texte (unicode)
- Format spécial
  - Analyse des documents et transformation en arbres
  - Navigation d'arbres
  - Manipulations d'arbres
  - Sérialiser des arbres en fichier texte



## Technologies

- Modèle DOM
  - Chargement complet en mémoire
  - DOM, JDOM, DOM4J
- SAX
  - "Streaming", pas de stockage
  - Apache Xerces (DOM + SAX) [Java, C++]
- JAXB
  - Passer d'un schéma à un jeu de classes de manière transparente ... (Java EE)
- Diverses bibliothèques : XStream



## Document Object Model

- Recommandation du W3C
- Indépendant du langage de programmation
- Support
  - Arbre XML
  - XPath, DTD, XML Schema
- Nœud
  - Document, Element, Attr, DocumentType, Notation, Entity, EntityReference, Processing Instruction, CharacterData (Text, Comment)



## JDOM

- Moins générique que DOM
- Pas d'analyseur fourni
  - utiliser DOM ou SAX
- Java avec collection
- Structures
  - Parent
    - Element, Document
  - Content
    - Comment, ProcessingInstruction, EntityReference, Text, DocType, Element



## Quelques classes/méthodes

- Document
  - getDescendants()
- Element
  - getContents(), get/setAttribute(), addContent()
- Namespace
- Filter



```
import org.jdom.*;
import org.jdom.input.*;

try {
    SAXBuilder b = new SAXBuilder();
    b.setValidation(true); // verification schema (DTD)
    // pour la validation XML schema
    b.setProperty("http://java.sun.com/xml/jaxp/
        properties/schemaLanguage",
        "http://www.w3.org/2001/XMLSchema");
    String msg = "Pas d'erreur";
    try {
        Document d = b.build(new File(args[0]));
    } catch (JDOMParseException e) {
        msg = e.getMessage();
    }
    System.out.println(msg);
} catch (Exception e) {
    System.err.println(e);
}
```

Exemple 1 :  
lecture de fichier XML

Fichier bien formé ?  
Validation optionnelle  
DTD par défaut



```
org.jdom, org.jdom.input, org.jom.ouput,
org.jdom.transform, javax.xml.transform

try {
    SAXBuilder b = new SAXBuilder();
    Document d = b.build(new File(args[0]));
    Transformer t =
        TransformerFactory.newInstance().newTransformer(
            new StreamSource(new File(args[1])));
    JDOMSource in = new JDOMSource(d);
    JDOMResult out = new JDOMResult();
    // t.setParameter(nom, valeur);
    t.transform(in, out);
    Document h = out.getDocument();
    XMLOutputter outputter = new XMLOutputter();
    outputter.output(h, System.out);
} catch (Exception e) {
    System.err.println(e);
    e.printStackTrace();
}
```

Exemple 2 :  
Transformation XSLT



## Simple API for XML

- Pas de stockage en mémoire comme les autres
  - Lecture en le moins de passes possibles
- SAX : reconstruire partiellement l'arbre
- Flux de caractères / événements
  - Méthodes *callbacks*

```
import org.xml.sax.*;
import org.xml.sax.helpers.*;
```



## Événements

- Début du document
- Balise de début
- Balise de fin
- Déclaration d'espace de nommage
- Espaces lus
- Données caractères lues
- Fin du document

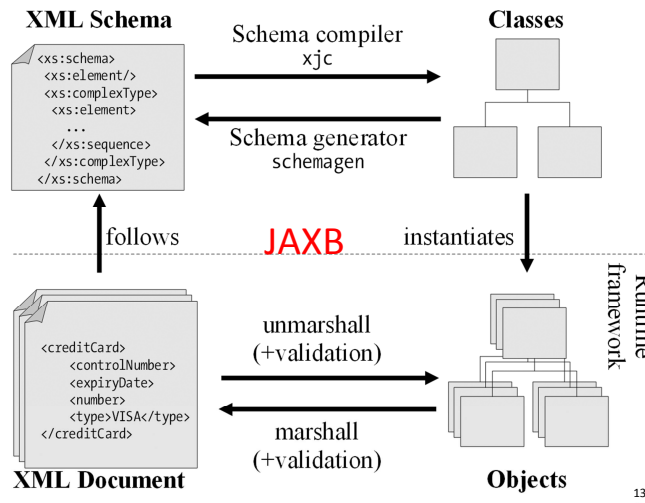


```

public class XMLSAXProject extends DefaultHandler {
    public void startDocument() {}
    public void endDocument() {}
    public void startElement( String uri, String localName,
        String qName, Attributes atts) {}
    public void endElement( String uri, String localName,
        String qName) {}
    public void ignorableWhitespace( char[] ch, int start,
        int length) {}
    public void characters( char[] ch, int start,
        int length) {}
    public void processingInstruction( String target,
        String data) {}

    public static void main(String[] args) {
        try {
            XMLSAXProject project = new XMLSAXProject();
            XMLReader read = XMLReaderFactory.createXMLReader();
            read.setContentHandler(project);
            read.parse(args[0]);
        } catch(Exception e) {
        }
    }
}

```



## XStream

- Initialiser

```
XStream xstream = new XStream();
```

- Utiliser des alias

```
xstream.alias("classe", Classe.class);
```

- Sérialiser

```
String s = xstream.toXML(objet);
```

- Désérialiser

```
Classe classe = (Classe)xstream.fromXML(string);
```

```

FileOutputStream fos = null;
XStream xstream = null;
try {
    fos = new FileOutputStream(name);
    xstream = new XStream();
    xstream.toXML(objects, fos);
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (fos!=null) fos.close();
}

```

Écrire

```

FileInputStream fis = null;
XStream xstream = null;
try {
    fis = new FileInputStream(name);
    xstream = new XStream();
    objects = (Composite)xstream.fromXML(fis);
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (fis!=null) fis.close();
}

```

Lire



## Bibliographie

- CSS2 Pratique du design web, 2<sup>ème</sup> édition, R Goetter, Eyrolles, 2005
- <http://openweb.eu.org>
- <http://www.w3school.com>
- <http://css.alsacreations.com>
- <http://www.gchagnon.fr/cours/xml/>
- An Introduction to XML and Web Technologies, A Moller, M Schwartzbach, Addison Wesley, 2006