

# Technologie Web

## CSS

**Alexandre Pauchet**

INSA Rouen - Département ASI

BO.B.RC.18, pauchet@insa-rouen.fr

# Plan

- 1 Introduction
- 2 Règles, éléments et propriétés
- 3 Propagation de règles
- 4 Positionnement
- 5 Exemple
- 6 XML et CSS

# Les CSS (1/12)

## Description

### Cascading Style Sheets Feuilles de style en cascade

- Un document en mode texte
- Utilisable par des documents HTML ou XML
- Présentation identique de tous les documents HTML ou XML
- Apporte la modularité du formatage
- Séparation du contenu et de la présentation
- 3 normes : CSS 1, CSS 2, CSS 3

# Les CSS (2/12)

## Inclusion d'une CSS

Trois possibilités d'inclusion :

- Directement dans les balises (à éviter)

```
<h2 style="color:red">Titre en rouge</h2>
```

- Définition de la CSS dans le fichier *via* la balise `<style>`

```
<head>  
  <style type="text/css">  
    déclaration des styles  
  </style>  
</head>
```

- Déclaration d'un lien vers la CSS via la balise `<link>`

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

# Les CSS (3/12)

## Déclaration d'une règle

```
sélecteur1, sélecteur 2 ... {  
    propriété1: valeur1;  
    ...  
    propriétéN: valeurN;  
}
```

- On peut mettre autant de couples propriété/valeur que voulu, séparés par des ;
- Commentaire : /\* **Commentaire** \*/

### Exemple

```
h1, h2 {color:red;}
```

# Les CSS (4/12)

## Déclaration d'une règle (suite)

- Sélecteur de type : nom de balise

```
h1 {color: red; background-color: yellow}
```

- Sélecteur universel : \*

- Sélecteur d'ID : #

```
#important {color: red}
```

Sélection de : `<p id="important">paragraphe</p>`

- Sélecteur de classe : .

```
.special {font-size: 20pt} ou h1.special {font-size: 60pt}
```

Sélection de : `<h1 class="special">Titre spécial</h1>`

### Remarque

Les id sont uniques sur une même page.

Les class s'appliquent à plusieurs balises.

# Les CSS (5/12)

## Déclaration d'une règle (suite)

- Sélecteur de descendant  
`p h2 {color: green} : "les h2 qui sont dans un p"`
- Sélecteur d'enfant (descendant direct)  
`p > h2 {font-size: 30pt} : "les h2 qui sont directement dans un p"`
- Sélecteur d'adjacent :  
`p ~ h2 {font-size: 10pt} : "les h2 qui sont après un p"`
- Sélecteur d'adjacent direct :  
`p + h2 {font-size: 10pt} : "les h2 qui sont directement après un p"`
- Sélecteur de pseudo-classe  
`a:visited {color: brown}`  
`p em:first-child {font-weight: bold}`  
`form:invalid {font-weight: bold} et`  
`input:required:valid {border-color: #008000;}`

# Les CSS (6/12)

## Déclaration d'une règle (suite et fin)

- Sélecteur de pseudo-élément "première lettre"

```
p:first-letter {font-size: 200%; font-weight: bold;
                float: left; text-transform: capitalize}
```

- Sélecteurs de pseudo-élément "avant" et "après"

- :before et :after génèrent du contenu avant et après un élément
- Le contenu généré est déclaré par la propriété content
- Les pseudo-éléments générés héritent du style de l'élément référence

```
dt:before { content: "->"; }
```

# Les CSS (6/12)

## Éléments et propriétés

- Propriétés (<http://www.w3schools.com/cssref/>)
  - polices de caractères :  
`font-size`, `font-style`, `font-family`, `font-weight`
  - paragraphes :  
`line-height`, `text-align`, `text-indent`, `text-transform`
  - blocs :  
`height`, `width`, `margin-right`, `margin-left`, `margin-top`, `margin-bottom`, `padding-right`, `border-style`, `border-top-width`, ...



# Les CSS (7/12)

## Cascade

Tous les styles, peu importe où ils sont définis, se **fusionnent** dans l'**ordre de chargement**, en une seule feuille de style avec un système d'**héritage** et peuvent s'**écraser**.

```
h1 {  
  color : red;  
  font-size : 18px;  
}  
h2 {  
  color : green;  
}
```

+

```
h1 {  
  color : blue;  
}  
h2 {  
  color : green;  
  font-size : 12px;  
}
```

=

```
h1 {  
  color : blue;  
  font-size : 18px;  
}  
h2 {  
  color : green;  
  font-size : 12px;  
}
```

# Les CSS (8/12)

## Cascade (suite)

Tous les styles applicables sont appliqués.

```
h1 {color : red;}  
p { background-color:grey; }  
.untitre { font-size : 64px; }
```

+

```
<p>  
  <h1 class= "untitre">Le titre qui a du style</h1>  
</p>
```

=

**Le titre qui a du style**

# Les CSS (9/12)

## Positionnement

- Fonctionnement par défaut :  
Dans le flux, les éléments sont positionnés les uns en dessous des autres
- Positionnement flottant (par rapport au bloc contenant) :  
`float: left;`
- Positionnement absolu (par rapport au conteneur, hors flux) :  
`position: absolute; top: x px; left: y px;`
- Positionnement fixe (absolu même en cas de scroll) :  
`position: fixed; top: x px; left: y px;`
- Positionnement relatif (par rapport à sa position dans le flux) :  
`position: relative; top: x px; left: y px;`

[http://www.w3schools.com/css/css\\_positioning.asp](http://www.w3schools.com/css/css_positioning.asp)

# Les CSS (10/12)

## Exemple d'utilisation

### Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple d'utilisation des CSS</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link href="Livres.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <dl>
      <dt>CSS pour les nuls</dt><dd>Yvon Tengrece</dd>
      <dt>XML pour pas mieux</dt><dd>Alain Stigateur</dd>
      <dt>XHTML pour <em>rien</em></dt><dd>Arnie Ka</dd>
    </dl>
    <p>Voici un petit texte juste pour <strong>montrer</strong> la différence
    avec une <span id="important">liste</span>. On remarquera que la largeur
    est fixée à <em>500px.</em></p>
    <p>et un second paragraphe derrière.</p>
    <div>Et un span dans un div pour tester br/i : <span class="i"><span
    class="br">i puis br (on favorise i)</span></span> et <span class="br"><
    span class="i">br puis i (on tient compte de i et br)</span></span>.</div>
  </body>
</html>
```

# Les CSS (11/12)

## Exemple d'utilisation

```
html { font-family: arial; font-size: 20px; font-style: normal; color: #000000;
       background: #cccccc; margin: 5px; width: 500px; }
p { text-align: justify }
p:first-letter { text-transform: capitalize }
p + p { margin-top: 10mm }
dl { margin-left: 50px; }
dt { font-style: italic; }
dt em { font-weight: bold; }
dd { color: #ff6600; }
#important { color: red }
.i>.br { font-weight: normal; font-style: italic; }
.br>.i { font-weight: bold; font-style: italic; }
```

# Les CSS (12/12)

## Exemple d'utilisation

### Sans CSS :

```
CSS pour les nuls  
Yvon Tengreçe  
XML pour pas mieux  
Alain Stigateur  
XHTML pour rien  
Arnie Ka
```

Voici un petit texte juste pour **montrer** la différence avec une liste. On remarquera que la largeur est fixée à 500px.  
et un second paragraphe derrière.

Et un span dans un div pour tester br/i : i puis br (on favorise i) et br puis i (on tient compte de i et br).

### Avec CSS :

```
CSS pour les nuls  
Yvon Tengreçe  
XML pour pas mieux  
Alain Stigateur  
XHTML pour rien  
Arnie Ka
```

Voici un petit texte juste pour **montrer** la différence avec une **liste**. On remarquera que la largeur est fixée à 500px.

Et un second paragraphe derrière.

Et un span dans un div pour tester br/i : *i puis br (on favorise i)* et **br puis i (on tient compte de i et br)**.

# XML et CSS (1/8)

## Les feuilles de style

- **Objectif** : séparation du contenu et de la présentation
  - ⇒ interopérabilité
    - Distribution à plusieurs médias (multiplateforme)
    - Personnalisation (adaptation de l'affichage aux groupes)
  - ⇒ réutilisabilité (traitement par des techniques encore inexistantes)
  - ⇒ faciliter la recherche d'informations
  
- Plusieurs formats :
  - CSS : *Cascading Style Sheets*
  - XSL : *eXtensible Stylesheet Language*
  - XSL-T : *eXtensible Stylesheet Language Transform*

# XML et CSS (2/8)

## Utilisation de CSS

### Principe

*XML [+ DTD] + CSS* ⇒ **Document formaté**

```
<?xml-stylesheet type="text/css" href="fichier.css" ?>
```

### Remarques :

- Toutes les balises définies dans un document XML / une DTD peuvent faire l'objet d'un traitement
- L'ordre des éléments (balises) dans un document XML est respecté ; il faut tenir compte de ces possibilités lorsque l'on rédige une CSS

# XML et CSS (3/8)

## Déclaration d'une règle

**Syntaxe :** `sélecteur[, sélecteur...]` {déclarations}

- sélecteur d'élément : `element`
- sélecteur d'id : `#id`
- sélecteur de classe : `.class`
- sélecteur universel : `*`
- Sélecteur de descendant : `element element`
- Sélecteur d'enfant (descendant direct) : `element>element`
- Sélecteur d'adjacent : `element1~element2`
- Sélecteur d'adjacent direct : `element+element`

## XML et CSS (4/8)

### Sélecteurs de paramètre

- `elem[att]` : sélectionne les éléments de type `elem` contenant l'attribut `att`, quelque soit la valeur de l'attribut.
- `elem[att=val]` : sélectionne les éléments de type `elem` dont l'attribut `att` vaut `val`.
- `elem[att~=val]` : sélectionne les éléments de type `elem` dont l'attribut liste `att` contient `val`.
- `elem[att^=val]` : sélectionne les éléments de type `elem` dont l'attribut `att` débute par `val`.
- `elem[att$=val]` : sélectionne les éléments de type `elem` dont l'attribut `att` se termine par `val`.
- `elem[att*=val]` : sélectionne les éléments de type `elem` dont l'attribut `att` contient la sous-chaîne `val`.

**Remarque** : il est possible de combiner plusieurs sélecteurs de paramètre  
`elem[sel1][sel2]...[seln]`

# XML et CSS (5/8)

## Éléments et propriétés

- Affichage d'un élément :
  - “display: block” ⇒ affichage comme élément *block*
  - “display: inline” ⇒ affichage comme élément *inline*
  - “display: none” ⇒ pas d'affichage
- Par défaut les élément sont considérés comme *inline*
- Propriétés identiques à celles du HTML
  - polices de caractères :  
`font-size`, `font-style`, `font-family`, `font-weight`
  - textes :  
`line-height`, `text-align`, `text-indent`, `text-transform`
  - blocs :  
`height`, `width`, `margin-right`, `margin-left`, `margin-top`, `margin-bottom`, `padding-right`, `padding-left`, `padding-top`, `padding-bottom`, `border-style`, `border-top-width`, `border-right-width`,  
...

# XML et CSS (6/8)

## CSS : génération de contenu

- Les pseudo-éléments `:before` et `:after` permettent de générer du contenu avant et après un élément
- Le contenu généré est déclaré par la propriété `content`
- Les pseudo-éléments générés héritent du style de l'élément référence

### Exemple

```
collection : before
{
  content : "Dans :";
  display : inline;
  font-style : italic;
}
```

# XML et CSS (7/8)

## Exemple

### Exemple

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE album [
  <!ELEMENT album (titre , auteur , contenu)>
  <!ELEMENT titre (#PCDATA)>
  <!ELEMENT auteur (#PCDATA)>
  <!ELEMENT contenu (piste+)>
  <!ELEMENT piste (#PCDATA)>
]>
<?xml-stylesheet type="text/css" href="styleAlbum.css"?>
<album>
  <titre>Le monde de la chatouille</titre>
  <auteur>Sacha Touille</auteur>
  <contenu>
    <piste>01 - Guiliguili</piste>
    <piste>02 - Gratouilli</piste>
  </contenu>
</album>
```

# XML et CSS (8/8)

## Exemple

### Exemple

```
* {  
  display: block;  
  padding-left: 10px;  
}  
  
album {  
  width: 500px;  
  background-color: teal;  
  font-family: arial ;  
  color: white;  
  font-size: 12pt ;  
}  
  
titre , auteur {  
  display: inline;  
  font-size: 16pt;  
  font-weight: bold;  
}  
  
auteur { color: red ; }  
auteur:before { content: " (" ; }  
auteur:after { content: ")" ; }
```

## Documentation et liens

- **HTML 5**

w2schools : <http://www.w3schools.com/html5/>

Toutes les balises :

[http://www.w3schools.com/html5/html5\\_reference.asp](http://www.w3schools.com/html5/html5_reference.asp)

- **CSS**

w2schools : <http://www.w3schools.com/html5/>

Balises :

[http://www.w3schools.com/html5/html5\\_reference.asp](http://www.w3schools.com/html5/html5_reference.asp)

- **Validation**

W3C (DTD based) : <http://validator.w3.org>

Validator.ne (non-DTD) : <http://html5.validator.nu>

L'indispensable Firebug pour le débogage !

- **Compatibilité navigateurs**

<http://caniuse.com>