INF2005 – Programmation web

CSS

Jacques Berger

Objectifs

L'utilité de CSS

La syntaxe de CSS

L'intégration à HTML

Prérequis

HTML

CSS

Cascading Style Sheets

Version courante : CSS 3

Le problème initial

Pollution du document HTML par la logique de présentation

Énormément de répétitions de balises

Difficile à maintenir et sujet aux oublis

Le problème initial

On devait répéter toutes les balises de style partout où l'on voulait ce style

```
<table>
  <tbody>
    <tr>
      <th><b><i><center>Titre</center></i></b></th>
      <th><b><i><center>Année</center></i></b></th>
    </tr>
    <tr>
      <td><font color="red">Under the Fall</font></td>
      <td><font color="red">2011</font></td>
    </tr>
  <tbody>
</table>
```

La solution

On enlève la présentation du document HTML

```
<table>
  <tbody>
    <tr>
      <th>Titre</th>
      <th>Année</th>
    </tr>
    <tr>
      <td>Under the Fall</td>
      <td>2011</td>
    </tr>
  </tbody>
</table>
```

La solution

On place la présentation dans la feuille de style

```
th {
    font-weight: bold;
    font-style: italic;
    text-align: center;
}

td {
    color: #FF0000;
}
```

Balises de style

À proscrire depuis HTML 4

```
basefont
center
font
s
strike
u
b
etc.
```

Attributs de style

Certains attributs d'élément sont également à proscrire

```
align
bgcolor
text
link
alink
vlink
background
etc.
```

HTML5

Depuis HTML5, les balises et attributs de style n'existent plus

Valider le document HTML aidera à les éviter

Syntaxe

```
Selecteur {propriete: valeur;}
```

Sélecteur
- Ce qui détermine sur quels éléments ce style s'appliquera

Propriété
- La propriété à modifier

Valeur
- La valeur à assigner à la propriété

Syntaxe

```
th {
    font-weight: bold;
    font-style: italic;
    text-align: center;
}


td {
    color: #FF0000;
}
```

Syntaxe

Les délimiteurs de commentaire :

```
/* ... */
```

Sélecteur

Le sélecteur peut être :

- Un élément
- Une séquence d'éléments
- Une classe
- Un id
- Une combinaison de toutes ces choses

Sélecteur

Un élément

```
p {
    color: #0000FF;
}
```

Utilisation avec HTML : automatique

```
<p>Ce texte sera bleu automatiquement.</p>
```

Sélecteur

Plusieurs éléments

```
h1, h2, h3 {
    font-size: 12px;
}
```

Utilisation avec HTML : automatique

```
<h1>Titre en taille 12</h1>
<h3>Également en taille 12</h3>
```

Sélecteur

Une séquence d'éléments

```
ol li {
    color: #000000;
}
```

Utilisation avec HTML : automatique, ne s'applique que si l'élément li est à l'intérieur de l'élément ol, peu importe la profondeur

Sélecteur

S'applique :

```
<ol>
   <li>Affiché en noir</li>
</ol>
```

Ne s'applique pas :

```
<ul>
   <li>Utilise la couleur par défaut</li>
</ul>
```

Sélecteur

Un élément selon son parent

```
p>a {
    text-decoration: none;
}
```

Utilisation avec HTML : automatique, ne s'applique que si l'élément parent de la balise a est p

```
<p>Voici un exemple où ça s'appliquera. <a
href="http://www.jberger.org">Cliquez ici.</a></p>
```

Sélecteur

Un élément avec une condition d'attribut

```
input[type="text"] {
    background: #FF0000;
}
```

Utilisation avec HTML : automatique, ne s'applique que sur les éléments input ayant un attribut type avec la valeur text

Sélecteur

Sélecteurs d'attributs

`a[href^="jberger"]` : s'applique aux éléments a qui possèdent l'attribut href et dont la valeur de l'attribut commence par jberger

`a[href$="org"]` : s'applique aux éléments a qui possèdent l'attribut href et dont la valeur de l'attribut termine par org

`a[href*="berger"]` : s'applique aux éléments a qui possèdent l'attribut href et dont la valeur de l'attribut contient berger

Sélecteur

Tous les éléments

```
* {
    background: #000000;
}
```

Utilisation avec HTML : automatique

Sélecteur

Un élément qui suit un autre : ~

`p ~ a` : s'applique aux éléments a qui apparaissent après un élément p

`div ~ p` : s'applique aux éléments p qui apparaissent après un élément div

id

Les éléments peuvent tous avoir un attribut id servant à identifier l'élément de façon unique dans le document

```
<div id="page">
    <p id="author">Jacques Berger</p>
</div>
```

Deux éléments ne peuvent avoir le même id, ils doivent tous être uniques

id

On s'en sert pour :

- Identifier l'élément dans le document
- Manipuler l'élément avec Javascript
- Associer un style à l'élément unique

id

Associer un style par id

```
#sidebar {
    background: #101010;
    font-size: 11px;
}
```

Utilisation avec HTML : attribut id

```
<div id="sidebar">
    …
</div>
```

Classe

2 utilités

Permettre à n'importe quel élément d'utiliser un style en particulier

Permettre d'avoir plusieurs styles pour une même balise

Classe

Plusieurs classes pour un élément

```
td.string {
    text-align: center;
}

td.money {
    text-align: right;
}
```

Utilisation avec HTML : attribut class

```
<td class="money">44.33 $</td>
```

Classe

Une classe pour plusieurs éléments

```
.logo {
    text-align: left;
    font-size: 36px;
    text-transform: uppercase;
}
```

Utilisation avec HTML : attribut class

```
<div class="logo">
    …
</div>

<span class="logo">INF2005 - programmation web</span>
```

Classe

On peut avoir une séquence de classes

```
.page .post {
    color: #000000;
}
```

Utilisation avec HTML : attribut class, s'applique lorsqu'un élément utilisant la classe post est situé à l'intérieur d'un élément utilisant la classe page, peu importe la profondeur

```
<div class="page">
    <div class="temp">
        <div class="post">
            …
        </div>
    </div>
</div>
```

Cascade

Les différents styles vont s'unifier à l'interprétation

```
h1,h2 {
    margin: 0px;
    font-weight: normal;
}

h1 {
    font-size: 44px;
}

h2 {
    font-size: 18px;
}
```

Cascade

Les propriétés vont également s'accumuler et s'écraser au fur et à mesure que les styles se précisent

Voir exemple

Cascade

La priorité des styles est calculée selon un algorithme complexe

Le poids du style vaut abcd où a, b, c, et d sont un seul chiffre

ex. a=1, b=1, c=0, d=0, le poids vaut 1100

Cascade

a vaut 1 si le style provient de l'attribut style, sinon a vaut 0

b vaut le nombre d'identifiants dans le sélecteur

c vaut le nombre de classes dans le sélecteur

d vaut le nombre d'éléments dans le sélecteur

Cascade

Exemples :

```
#content .money ul li
vaut 0112

.right div
vaut 0011

p
vaut 0001
```

Déclaration

On peut déclarer les styles à trois endroits :

- Dans un fichier CSS séparé
- Dans l'en-tête du document HTML
- Directement dans l'élément

Déclaration

Pour lier un document HTML à une feuille de style externe, ajoutez la ligne suivante dans l'entête du document (élément head) :

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

Déclaration

La feuille de style dans l'en-tête du document html (élément head) :

```
<style type="text/css">
    …
</style>
```

Déclaration

Méthode inline

On utilise l'attribut style de l'élément

```
<p style="color:red">Texte rouge</p>
```

Déclaration

Méthode recommandée : fichier externe

Méthode à éviter : inline

Codes de couleur

Il existe 17 noms standards pour nommer les couleurs

red, blue, green, gray, grey, etc.

Pour les autres, on utilise l'encodage RGB

Codes de couleur

RGB
    Red-Green-Blue

Encodé en hexadécimal, sur 24 bits

#FFFFFF = blanc
#000000 = noir
#FF0000 = rouge
#00FF00 = vert
#0000FF = bleu

Unités de mesure

% : pourcentage
cm : centimètres
em : taille actuelle de la police
in : pouces
mm : millimètres
px : pixels

Etc.

Propriétés

background

Définit le fond d'un conteneur; peut être une couleur ou une (ou plusieurs) image

```
background: #E0D6CC;
```

Propriétés

font-family

Définit la police de caractères à utiliser;
on spécifie plusieurs polices en ordre de priorité;
on termine par une famille générique

Les noms de police avec des espaces doivent être entre guillemets

```
font-family: Arial, Helvetica, sans-serif;
```

Propriétés

font-size

Définit la taille de la police de caractères

```
font-size: 12px;
```

Propriétés

font-weight

Définit l'épaisseur du texte

```
font-weight: bold;
```

Propriétés

font-style

Définit le style à appliquer au texte (oblique, italique, normal)

```
font-style: italic;
```

Propriétés

color

Définit la couleur du texte

```
color: #FFFFFF;
```

Propriétés

line-height

Définit la hauteur d'une ligne de texte

```
line-height: 240%;
```

Propriétés

text-align

Définit l'alignement du texte

```
text-align: justify;
```

Propriétés

text-decoration

Définit les artifices qu'on ajoute au texte (souligné, surligné, barré)

```
text-decoration: none;
```

Propriétés

text-transform

Définit quelle transformation appliquer au texte (lowercase, uppercase, capitalize)

```
text-transform: lowercase;
```

Propriétés

width

Définit la largeur d'un élément

```
width: 640px;
```

Propriétés

height

Définit la hauteur d'un élément

```
height: 160px;
```

Propriétés

list-style

Définit le graphique à utiliser pour les listes non-ordonnées (peut être une image)

```
list-style: none;
```

Propriétés

display

Définit le type de boîte que l'élément va générer; peut également servir à cacher l'élément

```
display: block;
```

Propriétés

float

Définit l'endroit où l'élément va flotter

Pour bien comprendre cette propriété, l'idéal c'est de l'essayer!

```
float: left;
```

Propriétés

visibility

Sert à cacher ou afficher un élément

```
visibility: hidden;
```

Propriétés

Il existe plusieurs autres propriétés

Consultez la référence!

URL

Pour utiliser une image comme fond d'un conteneur

```
div.bg {
    background: url(images/img03.jpg) no-repeat left top;
}
```

Boîtes

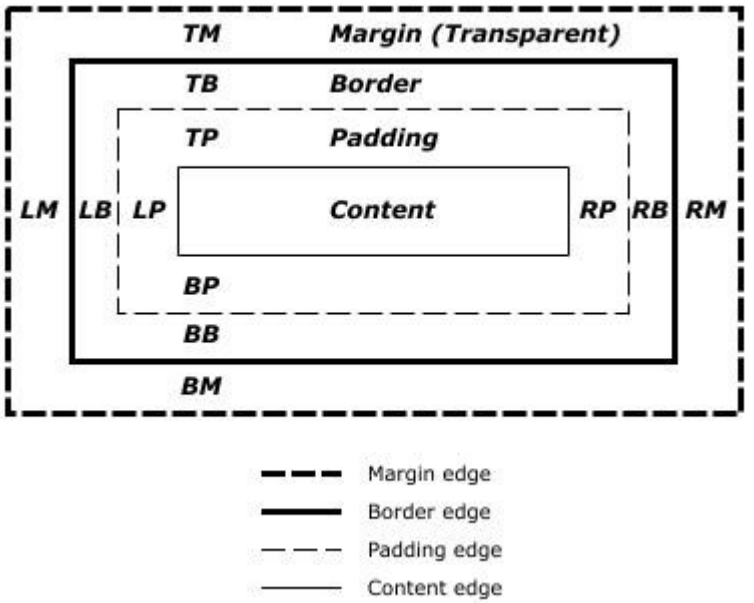

Image tirée de http://www.w3.org/TR/CSS2/box.html

Boîtes

Propriétés

- margin
- padding

Valeurs : haut, droite, bas, gauche

```
margin: 0px;              vaut   margin: 0px 0px 0px 0px;
margin: 0px 10px;         vaut   margin: 0px 10px 0px 10px;
margin: 0px 10px 20px;    vaut   margin: 0px 10px 20px 10px;
```

Boîtes

```
padding-top
padding-right
padding-bottom
padding-left

margin-top
margin-right
margin-bottom
margin-left
```

Boîtes

La propriété border fonctionne différemment

On doit lui spécifier un style

```
border: 2px solid black;
```

Boîtes

Propriétés pour avoir une boîte avec les coins arrondis

```
border-radius

border-top-right-radius
border-bottom-right-radius
border-top-left-radius
border-bottom-left-radius
```

Boîtes

Le border d'une boîte peut être une image

```
border-image

border-image-source
border-image-slice
border-image-width
border-image-outset
border-image-stretch
```

Pseudo-classes

On peut ajouter une pseudo-classe à un sélecteur CSS pour y spécifier un contexte d'application

```
selecteur:pseudo-classe {propriete: valeur;}

Ex.
a:hover {text-decoration: none;}
```

Pseudo-classes

:hover : Sélectionne l'élément lorsque la souris est au-dessus

:focus : Sélectionne l'élément input qui détient le focus

Pseudo-classes

:link : Sélectionne les liens non visités

:visited : Sélectionne les liens visités

:active : Sélectionne le lien actif

Pseudo-classes

:root : Sélectionne l'élément html

:empty : Sélectionne les éléments p vides

:enabled : Sélectionne les éléments activés

Pseudo-classes

:disabled : Sélectionne les éléments input désactivés

:checked : Sélectionne les cases cochées

:not(selecteur) : Inverseur sur un sélecteur

Pseudo-classes

Il existe d'autres pseudo-classes, notamment pour la manipulation des éléments p

Plus loin...

Introduction au CSS

http://www.w3.org/TR/css3-roadmap/