

# Cours Web n°2

## Introduction à CSS

Sandrine-Dominique Gouraud (gouraud@lri.fr)  
Pierre Senellart (pierre@senellart.com)



Semaine du 26 septembre 2005

# Plan du cours

- 1 CSS et XHTML
- 2 Sélecteurs de CSS
- 3 Propriétés CSS simples
- 4 Références
- 5 Application

- CSS : **C**ascading **S**tyle**S**heets
- Recommendation du W3C
- Plusieurs niveaux : CSS1 (1996), CSS2 (1998), CSS2.1 (2005), CSS3 (en cours)
- Support par les navigateurs très inégal, jamais complet (en particulier, IE6 a plusieurs limitations importantes)  $\implies$  on ne parlera que de ce qui marche bien dans l'ensemble des navigateurs courants.

## Principe

- (X)HTML décrit la **structure** et le **contenu**
- CSS décrit la **mise en forme** :
  - ▶ du texte (ce cours)
  - ▶ des **boîtes** les unes par rapport aux autres (cours suivant)

- Manière le plus simple d'utiliser les CSS.
- Rajouter un attribut `style` sur les balises XHTML.
- On peut utiliser `span` si on a besoin d'une balise supplémentaire.
- Encombre le code XHTML avec des indications de mise en forme : ce n'est pas ce qu'on veut !

## Exemples

- Ce mot en `<emph style="color: red;">emphase</emph>` est aussi en rouge.
- `<span style="text-decoration: underline">Cette expression de plusieurs mots</span>` est soulignée.

- Mettre la feuille de style CSS dans un fichier à part (en général, on utilise l'extension `.css`).
- Permet d'utiliser la même feuille de style pour plusieurs pages Web.
- Rajouter une balise `<link>` dont l'attribut `rel` est positionné à `stylesheet` dans l'en-tête du document.

## Exemple

```
<!DOCTYPE ... >
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    ...
    <link rel="stylesheet" href="feuille.css"
          type="text/css" />
  </head>
  <body> ... </body>
</html>
```

- On veut parfois rajouter encore plus de structure et de sémantique à un document XHTML.
- On utilise l'attribut `class` sur n'importe quelle balise (ou, à défaut, sur une balise `span`).
- Après, on peut utiliser CSS pour appliquer une mise en forme commune à tout ce qui fait partie d'une `class` particulière.

### Exemple (Mettre en bleu italique les noms de personnes)

- XHTML

```
<p>Je voudrais particulièremment remercier  
<span class="personne">Madame Machin</span>  
et <span class="personne">Monsieur Bidule</span>.</p>
```

- CSS

```
.personne {  
  color: blue;  
  font-style: italic;  
}
```

- Ensemble de règles de la forme :  

```
sélecteur {  
    propriété: valeur;  
    ...  
}
```
- *sélecteur* : indique à quelles parties du documents la règle s'applique (cf partie 2 de ce cours).
- *propriété* : propriété spécifique de mise en forme à modifier (cf partie 3 de ce cours et cours suivant).
- *valeur* : son sens dépend de la propriété.
- Il faut **valider** les feuilles de style avec un validateur approprié, cf <http://jigsaw.w3.org/css-validator/validator-uri.html>
- Commentaires entre `/*` et `*/`.

- Plusieurs feuilles de style peuvent s'appliquer simultanément :
  - ▶ Plusieurs balises `<link rel="stylesheet">`
  - ▶ Directive `@import` d'une feuille de style  
`@import url(feuille_annexe.css);`
  - ▶ Feuille de style de l'utilisateur (Mozilla, Opera...)
- Au sein même d'une feuille de style, plusieurs règles peuvent être en conflit.
- **Cascade** : mécanismes gérant ces conflits.

## Cascade

- Si `!important` est précisé après la valeur, la règle sera prioritaire.
- Sinon, plus une règle est spécifique, plus elle est prioritaire.
- Sinon, la dernière règle s'applique.

# Plan du cours

- 1 CSS et XHTML
- 2 Sélecteurs de CSS**
- 3 Propriétés CSS simples
- 4 Références
- 5 Application

**Sélecteur simple** : nom d'une balise.

**Sélecteur multiple** : plusieurs sélecteurs séparés par des virgules.

**Sélecteur universel** : '\*', sélectionne tout.

## Exemples

- `ul { color: blue; }` met l'ensemble du contenu des listes non ordonnées en bleu.
- `h1,h2,h3,h4,h5,h6 { color: red; }` met l'ensemble des titres de rubrique en rouge.
- `* { color: black; }` met tout en noir. Dans ce cas précis, on préférera une règle `body { color: black; }`.

**Sélecteur de classe** : nom d'une classe, préfixée d'un '.', tel qu'il apparaît dans un attribut `class` d'une balise XHTML.

## Exemples

- `.personne { font-face: bold; }` met l'ensemble des balises de classe `personne` en gras.
- `p.comment { font-style: italic; }` met l'ensemble des balises `p` de classe `comment` en italique.

**Identifiant** : défini par l'attribut `id` d'une balise XHTML. Similaire aux classes, mais il ne peut y avoir qu'**une seule** balise ayant un `id` donné dans tout le document XHTML.

**Sélecteur d'identifiant** : nom d'une classe, préfixée d'un '#', tel qu'il apparaît dans un attribut `id` d'une balise XHTML.

## Exemples

- `#introduction { font-size: 120%; }` met la balise d'identifiant `introduction` en plus gros.
- `p#introduction { font-size: 120%; }` met la balise `p` d'identifiant `introduction` en plus gros.

**Sélecteur contextuel** : 2 sélecteurs ou plus séparés par des espaces.  $A B$  sélectionne les  $B$  seulement s'ils sont contenus dans des  $A$ .

## Exemples

- `h1 em { color: blue; }` met les mots en emphase à l'intérieur d'un `h1` en bleu.
- `ul ol, ol ul, ul ul, ol ol { font-size: 80%; }` diminue la taille du texte des listes imbriquées.

**Pseudo-élément** : permet de mettre une règle sur quelque chose qui n'est pas réellement une balise XHTML.

## Exemples

- `p:first-line { font-face: bold; }` met la première ligne de chaque paragraphe en gras.
- `p:first-letter { font-face: bold; }` met la première lettre de chaque paragraphe en gras.

**Pseudo-classes** : permet de faire une sélection d'un élément uniquement dans certains contextes.

**a:link** : les a qui sont des liens.

**a:visited** : les a qui sont des liens vers des pages qui ont déjà été visitées.

**a:hover** : les a qui sont des liens qu'on est en train de désigner (i.e. la souris est en train de passer dessus).

**a:active** : les a qui sont des liens qu'on est en train d'activer (i.e. de cliquer).

### Remarque

**Attention!** Toujours définir ces pseudo-classes dans ce sens, à cause des règles de cascade.

# Plan du cours

- 1 CSS et XHTML
- 2 Sélecteurs de CSS
- 3 Propriétés CSS simples**
- 4 Références
- 5 Application

Les propriétés admettent différentes unités de mesure se répartissant suivant trois types de valeurs :

- les valeurs fournies sous forme de pourcentage (par rapport à la valeur courante)
- les valeurs relatives :
  - `em` la valeur de la propriété `font-size` de la police utilisée (largeur du caractère `m`)
  - `ex` la hauteur du caractère `x` dans la police utilisée
- les valeurs relatives à l'écran
  - `px` le nombre de pixels
- les valeurs absolues (N'ont aucun sens pour une page Web destinée à s'afficher sur un écran d'ordinateur !)
  - `mm` millimètre
  - `cm` centimètre (1cm= 10mm)
  - `in` inch (1in= 2.54cm)
  - `pt` point (1pt= 1/72in)
  - `pc` pica (1pc= 12pt)

- Il existe différents moyens pour contrôler les polices de caractères.
- Les principales propriétés (`font-family`, `font-size`, `font-weight` et `line-height`) seront responsables de l'aspect de votre texte.

Exemple (Règle définissant que tous les paragraphes sont en Times 13pt avec un espace interligne de 10pt)

```
p {  
  font-family: times;  
  font-size: 13pt;  
  font-weight: bold;  
  line-height: 10pt;  
}
```

`font-family` spécifie la police ou famille de police (serif, sans-serif, cursive, fantasy, monospace) à utiliser par le navigateur. Il est recommandé de déclarer une famille générique après les polices (séparer par une virgule) pour obliger le navigateur si la police n'est pas trouvée à lui substituer une police similaire.

`font-size` définit la taille de la police (exprimée en points, pourcentage, taille absolue ou relative, pixels).

`font-style` définit le style de police à utiliser à savoir `italic`, `oblique` ou `normal`.

`font-variant` définit la casse des caractères : `normal` ou `small-caps` (petites majuscules).

`font-weight` définit la graisse des caractères : `normal`, `bold` (gras).

`line-height` définit l'espace interligne ou la hauteur intrinsèque de tout élément non remplacé. Une bonne règle pour le texte dans le corps est de régler l'espace interligne à 1.5 fois la taille de la police (1.5em ou 150%).

`text-decoration` définit l'ornement du texte : `none`, `underline` (souligné), `overline` (surligné), `blink` (clignotant) et `line-through` (barré).

## Exemple

Un morceau de test peut être surligné, souligné ou encore ~~barré~~. Un mot peut être en **gras** ou en *italique*, sans sérification ou encore écrit façon machine à écrire.

- Les propriétés `letter-spacing` et `word-spacing` permettent de définir l'espace entre les lettres d'un mot et les mots dans une phrase.
- Cet espace peut être positif ou négatif selon si on cherche à éloigner ou rapprocher ces éléments.
- Comme les couleurs, ces propriétés doivent être utilisées avec **discernement**.

### Exemple

Les mots de cette phrase sont  
espacés de 1cm et les lettres  
de 0.25em.

- `text-align` définit la justification et l'alignement d'un texte : `left`, `right`, `center` et `justify` (aligne le texte sur les marges gauche et droite).
- `vertical-align` définit l'alignement vertical de la ligne de base d'un élément en fonction de la propriété `line-height` : `super` (texte en exposant), `sub` (texte en indice), `baseline` (normal), etc.
- `text-indent` permet de décaler le début de la première ligne de texte d'une valeur fixe ou proportionnelle à la valeur du paragraphe.

`list-style-image` définit l'image qui sera employée comme marqueur d'item de liste. Quand l'image est indisponible, elle remplace le marqueur produit par la propriété `list-style-type`

`list-style-type` définit l'apparence d'une liste numérotée ou non : `disc` (cercle plein), `circle` (cercle vide), `square` (carré), `decimal` (nombres), `upper-alpha` ou `lower-alpha` (lettres de l'alphabet), `upper-roman` ou `lower-roman` (chiffres romains) et `none`

## Exemple

- petit cercle plein
- petit cercle vide
- petit carré plein
- Aucun item

Les propriétés qui s'appliquent aux bordures ont entre 1 et 4 valeurs :

- Si une seule valeur est spécifiée : elle s'applique à **tous** les côtés.
- Si 2 valeurs sont spécifiées :
  - ▶ la première s'applique aux bordures horizontales
  - ▶ la seconde aux bordures verticales
- Si 3 valeurs sont spécifiées :
  - ▶ la première s'applique à la bordure du haut
  - ▶ la deuxième aux bordures verticales
  - ▶ la troisième à la bordure du bas
- Si 4 valeurs sont spécifiées : celles-ci s'appliquent respectivement aux bordures du haut, de droite, du bas et de gauche.

`border-style` définit le style des bordures : none, dotted, dashed, solid, double, groove, ridge, inset ou outset.

`border-color` définit la couleur des côtés de la bordure.

`border-width` spécifie l'épaisseur des bordures.

**Important :** sur une table, définir également `table { border-collapse: collapse; }` afin que les bordures de la table fusionnent.

- La couleur doit être utilisée avec **précaution**.
- Son rôle est de différencier les éléments d'un texte en jouant sur la notion de contraste.
- Il existe différentes manières de spécifier les valeurs d'une couleur :
  - ▶ En indiquant le nom d'une couleur prédéfinie parmi `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `lime`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white` et `yellow`
  - ▶ En utilisant le code RGB pour (Red, Green, Blue) avec des valeurs de 0 à 255
  - ▶ En utilisant la valeur hexadécimale
- Il existe des outils pour faciliter la définition de nouvelles couleurs :
  - ▶ ColorBlender  
<http://www.meyerweb.com/eric/tools/color-blend/>
  - ▶ Color Schemer  
<http://www.colorschemer.com/online.html>
  - ▶ la boîte à couleurs  
<http://pourpre.com/colorbox/index.php>

## Exemple

Acceptable sauf sur fond rouge

Acceptable sauf sur fond bleu

## Contre-exemple

Jaune sur blanc est typiquement à proscrire

Marron sur noir aussi.

## Exemple (Expression du bleu)

- Utiliser le mot clé blue
- Utiliser la commande rgb avec des valeurs absolues `rgb(0,0,255)` ou proportionnelles `rgb(0%,0%,100%)`
- Utiliser le code hexadécimal `#0000FF`

- La propriété *couleur* s'applique généralement au texte (`color`) ou à l'arrière-plan (`background-color`).
- La propriété `background-image` définit l'image d'arrière-plan d'un ou plusieurs éléments ou encore de tous les éléments. Il est fortement recommandé de spécifier aussi une couleur normale d'arrière-plan, qui sera utilisé en remplacement.

## Exemple

```
background-image: url("mes images/monImage.jpg");  
background-color: white;
```

# Plan du cours

- 1 CSS et XHTML
- 2 Sélecteurs de CSS
- 3 Propriétés CSS simples
- 4 Références**
- 5 Application

- Les spécifications de CSS :
  - ▶ <http://www.w3.org/TR/REC-CSS1>
  - ▶ <http://www.w3.org/TR/REC-CSS2/>
- Des sites Web :
  - ▶ <http://www.w3schools.com/css/default.asp>
  - ▶ <http://www.aidenet.com/css/>
- Des livres :
  - ▶ *HTML et XHTML : La Référence*, O'Reilly
  - ▶ *CSS : La Référence*, O'Reilly

# Plan du cours

- 1 CSS et XHTML
- 2 Sélecteurs de CSS
- 3 Propriétés CSS simples
- 4 Références
- 5 Application**

Compléter votre fichier *essai.html* et créer une feuille de style correspondante afin de le rendre identique au modèle de la copie d'écran disponible sur la page du cours.

## Règles du jeu

- Interdiction de rajouter autre chose au XHTML que :
  - ▶ Une balise `link`.
  - ▶ Des attributs `class` ou `id` sur des balises existants.
  - ▶ Des balises `<span>` avec `class` ou `id`.
- XHTML et CSS doivent être valides !