

Technologies du Web

Mise en forme et mise en page avec CSS

Pierre Senellart (pierre.senellart@telecom-paristech.fr)



Mastère spécialisé *Management et nouvelles technologies*,
5 octobre 2009

Plan du cours

- 1 CSS et HTML
 - Introduction
 - Intégration à HTML
 - Intégration à CSS
- 2 Sélecteurs de CSS
- 3 Mise en forme
- 4 Mise en page
- 5 Fonctionnalités mal supportées
- 6 Outils et Références

Intérêt

- CSS : **C**ascading **S**tyl**S**heets
- Recommandation 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 et IE7 ont plusieurs limitations importantes) \implies on se concentrera sur ce qui marche bien dans l'ensemble des navigateurs courants.

Principe

- HTML décrit la **structure** et le **contenu**
- CSS décrit :
 - ▶ la **mise en forme** du texte
 - ▶ la **mise en page** des **boîtes** les unes par rapport aux autres

Styles en ligne

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

Exemples

- Ce mot en `<em style="color: red;">emphase` est aussi en rouge.

```
<span style="text-decoration: underline">Cette expression de  
plusieurs mots</span> est soulignée.
```

Styles en en-tête

- Intégration des propriétés de style à l'**en-tête** de la page avec la balise `<style>`
- Utilisation des **sélecteurs** pour définir à quels élément les propriétés s'appliquent
- Inconvénients : mélange HTML et CSS dans le même document, impossible de réutiliser les propriétés CSS dans plusieurs documents

Exemple

```
<!DOCTYPE ... >
<html>
  <head> ...
    <style type="text/css">
      em { color: red; }
    </style>
  </head>
  <body> ... </body>
</html>
```

Feuille de styles liée

- 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.
- Possibilité d'ajouter `media="screen"` ou `media="print"`, etc., pour choisir différentes feuilles de style suivant le mode d'affichage.

Exemple

```
<!DOCTYPE ... >
<html>
  <head> ...
    <link rel="stylesheet" href="feuille.css" type="text/css">
  </head>
  <body> ... </body>
</html>
```

Classes

- On veut parfois rajouter encore plus de structure et de sémantique à un document HTML.
- On utilise l'attribut `class` sur n'importe quelle balise (ou, à défaut, sur une balise ``).
- 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)

• HTML

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

• CSS

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

Syntaxe de CSS

- Ensemble de règles de la forme :

```
sélecteur {  
    propriété: valeur;  
}
```

- **sélecteur** : indique à quelles parties du documents la règle s'applique
- **propriété** : propriété spécifique de mise en forme à modifier
- **valeur** : son sens dépend de la propriété.
- Les feuilles de style peuvent être **validées** avec un validateur approprié, cf. <http://jigsaw.w3.org/css-validator/>
- Commentaires entre /* et */.

Cascade

- 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 HTML
- 2 Sélecteurs de CSS
- 3 Mise en forme
- 4 Mise en page
- 5 Fonctionnalités mal supportées
- 6 Outils et Références
- 7 Application

Sélecteurs simples, multiples, universel

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; }`.

Sélecteurs de classes

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

Exemples

- `.personne { font-weight: 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.

Sélecteurs d'identifiants

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

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

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.

Sélecteurs contextuels

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-éléments

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

Exemples

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

Pseudo-classes

- **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. (**Las Vegas (Forest) Has Animals**).

Plan du cours

- 1 CSS et HTML
- 2 Sélecteurs de CSS
- 3 Mise en forme**
- 4 Mise en page
- 5 Fonctionnalités mal supportées
- 6 Outils et Références
- 7 Application

Propriétés de longueur

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

- 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 (hauteur du bloc dans lequel s'inscrit naturellement un caractère)
 - `ex` la hauteur du caractère `x` dans la police utilisée
- les valeurs relatives à l'écran (à éviter pour avoir une conception de la page indépendante de la résolution !)
 - `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`, `cm`, `in`, `pt`, `pc`

Polices

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

Exemple

```
p {  
  font-family: "Times New Roman";  
  font-size: 130%;  
  font-weight: bold;  
  line-height: 150%;  
}
```

`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 (la plus part du temps exprimée en pourcentage de la taille courante).

`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 1.2 fois la taille de la police (1.2em ou 120%).

`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*, avec empassements ou encore écrit façon machine à écrire.

Espacement

- Les propriétés `letter-spacing` et `word-spacing` permettent de définir l'espacement entre les lettres d'un mot et les mots dans une phrase.
- Cet espacement 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.

Alignement et indentation

`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.

Propriétés des listes

`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

Couleurs

- La couleur doit être utilisée avec **discernement**.
- 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

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

Plan du cours

- 1 CSS et HTML
- 2 Sélecteurs de CSS
- 3 Mise en forme
- 4 Mise en page
 - Modèle de boîtes de CSS
 - Flottement et positionnement
 - Tables
- 5 Fonctionnalités mal supportées
- 6 Outils et Références

Blocs et éléments en ligne

- Il existe deux sortes d'éléments HTML :
 - ▶ Les **blocs** : `<p>`, `<h1>`, `` ...
`<div>` est un bloc générique.
 - ▶ Les éléments **en ligne**, qui doivent être placés à l'intérieurs de blocs :
`<a>`, ``, `` ...
`` est un élément en ligne générique.
- On s'intéresse ici principalement aux **blocs** et à la manière dont les placer les uns par rapport aux autres.

Boîte CSS

margin (marge)

border (bordure)

padding (espacement)



Bordures des boîtes

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.

Largeur, Hauteur

- `width` : Largeur du **contenu** de l'élément : *longueur* ou `auto`. Par défaut à `auto`, prend la taille maximale disponible.
- `height` : Hauteur du **contenu** de l'élément : *longueur* ou `auto`. Par défaut à `auto`, s'adapte à la taille du contenu.

Remarque

`width` et `height` peuvent être également utilisés pour les éléments **en ligne remplacés** (comme ``).

Débordement

- Si la hauteur du contenu est contrainte, le contenu peut déborder d'un bloc...
- La propriété `overflow` contrôle le comportement :
 - ▶ `visible` : le contenu déborde de l'élément.
 - ▶ `hidden` : le contenu est tronqué.
 - ▶ `scroll` : des barres de défilement sont ajoutées.
 - ▶ `auto` : des barres de défilement sont ajoutées quand c'est nécessaire.
- Par défaut : `visible`.

Remarque

Internet Explorer 6 ne gère pas `overflow: visible;`

Marges

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`
- `margin` permet de définir d'un seul coup ces quatre propriétés, dans l'ordre *top right bottom left*.
- Par défaut : 0.
- Une marge peut être définie à `auto`, elle est alors automatiquement calculée à partir des autres valeurs.

Espacement

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`
- `padding` permet de définir d'un seul coup ces quatre propriétés, dans l'ordre *top right bottom left*.
- Par défaut : 0.

Cas particulier : centrage

Pour centrer un bloc horizontalement :

- Mettre `margin-left` et `margin-right` à `auto` .
- Donner une valeur à `width` différente de `auto` .

Flottement et positionnement

- Le **flottement** et le **positionnement** sont des outils permettant une mise en page complexe : mise en forme en colonnes, chevauchement d'éléments, etc.
- Le positionnement consiste à préciser où doivent apparaître des éléments de manière relative, par rapport à un autre élément ou encore par rapport à la fenêtre du navigateur.
- Le flottement n'est pas vraiment un positionnement : les éléments dits flottants sont pris dans le flux et les autres éléments les contournent.

Flottement

- Le flottement est défini par la propriété `float` qui peut prendre les valeurs :
 - ▶ `left` : l'élément sera contourné par la droite.
 - ▶ `right` : l'élément sera contourné par la gauche
 - ▶ `none` : valeur par défaut, sert principalement pour écraser un style existant.
- Tout bloc (image, texte, etc.) peut être défini comme un élément flottant.
- Il existe une dizaine de règles précises qui gouvernent le comportement des éléments flottants.

Dégagement

- La propriété `clear` indique quels côtés d'un élément ne doivent pas être adjacents à un élément flottant. Elle peut prendre les valeurs :
 - ▶ `left` : l'élément se positionne en-dessous de tout élément flottant à gauche.
 - ▶ `right` : l'élément se positionne en-dessous de tout élément flottant à droite.
 - ▶ `both` : l'élément se positionne en-dessous de tout élément flottant
 - ▶ `none` : valeur par défaut, sert principalement pour écraser un style existant.

Positionnement

- Le positionnement est précisé par la propriété `position`.
- Il existe 4 modes de positionnement : `relative`, `absolute`, `fixed` et `static`.
- Chaque type a ses propres règles.
- Les boîtes peuvent utiliser différents schémas et interférer entre elles.

Positionnement statique et relatif

- Le positionnement `static` est le comportement par défaut :
 - ▶ l'élément est une boîte rectangulaire faisant partie du flux i.e. l'ordre des déclarations contenues dans le **code source est respecté**.
- Le positionnement `relative` permet de **décaler** un élément d'une certaine distance.

Positionnement absolu et fixe

- Le positionnement `absolute` permet de retirer totalement un élément du flux et de le positionner par rapport à son conteneur (plus précisément, par rapport au conteneur le plus proche ayant un positionnement relatif).
- Le positionnement `fixed` permet de retirer totalement un élément du flux et de le positionner par rapport à la fenêtre d'affichage.
Attention : Ne fonctionne pas avec Internet Explorer 6.

Remarque

Pour un positionnement absolu par rapport à la page (cas le plus fréquent), on utilisera `body { position: relative; }`.

Décalage

- Une fois le type de positionnement choisi, on peut spécifier les propriétés de décalage : `top`, `right`, `bottom` et `left`.
- Le décalage se décrit à partir du bord le plus proche du bloc conteneur.
- Leur valeur peut être exprimée via une longueur, un pourcentage (par rapport à la taille du bloc conteneur) ou être choisie automatiquement `auto`.

Visibilité

- La visibilité d'un élément peut être complètement contrôlée en utilisant la propriété `visibility` :
 - ▶ `visible` permet de rendre visible l'élément
 - ▶ `hidden` permet de rendre invisible un élément ; la mise en page continue à le prendre en compte

Tables et CSS

- La place de la légende peut être précisée par la propriété `caption-side` : elle peut être au-dessus `top` ou en-dessous `bottom` de la table. (**Attention** : Ne fonctionne pas avec Internet Explorer 6/7).
- Les bordures des cellules peuvent :
 - ▶ fusionner `border-collapse: collapse`
 - ▶ être séparées `border-collapse: separate`. La propriété `border-spacing` précise alors l'espacement entre deux bordures (**Attention** : Ne fonctionne pas avec Internet Explorer 6/7).
- Les cellules vides identifiées par la propriété `empty-cells` peuvent apparaître `show` ou non `hide`. (**Attention** : Ne fonctionne pas avec Internet Explorer 6/7).
- On peut régler largeur et hauteur des cellules avec `width` et `height`.

Plan du cours

- 1 CSS et HTML
- 2 Sélecteurs de CSS
- 3 Mise en forme
- 4 Mise en page
- 5 Fonctionnalités mal supportées**
- 6 Outils et Références
- 7 Application

En CSS2.1

- `p>em` sélectionne les `` qui sont directement à l'intérieur d'un `<p>` (pas IE6)
- `li+li` sélectionne les `` qui suivent juste un autre (pas IE6)
- `*:lang(fr)` sélectionne les éléments dont la langue est définie comme le français (avec un attribut `lang=`, pas IE6 et IE7)
- `em:after {content: "!"}` ajoute un « ! » après tous les `` (`*:before` existe aussi, pas IE6 et IE7)
- Valeur `inherit` indique que la valeur est héritée des éléments de la hiérarchie (pas IE)
- `max-width` / `min-height` / etc. permettent de donner des hauteurs et largeurs minimales et maximales aux blocs (pas IE6)
- `display: table-cell;` (et `table`, `table-row`...) permet de mettre en page n'importe quels éléments comme des tables (pas IE6 et IE7)
- Compteurs automatiques (pas IE6 et IE7), et beaucoup d'autres choses !...

En CSS3

- Plus de noms de couleurs (marche **presque dans IE!**)
- Différentes feuilles de style suivant la résolution de l'écran, etc. (**seulement Opera**)
- Sélecteurs suivant la valeur d'un attribut (ex. `a[href$=".pdf"]` , **pas IE6**)
- Sélectionne les n^{es} , $2n^{\text{es}}$, etc., éléments à l'intérieur d'un autre élément (**quasiment pas supporté**)
- Et énormément d'autres choses, toujours en développement

Plan du cours

- 1 CSS et HTML
- 2 Sélecteurs de CSS
- 3 Mise en forme
- 4 Mise en page
- 5 Fonctionnalités mal supportées
- 6 Outils et Références**
- 7 Application

Outils

- N'importe quel éditeur de texte
- N'importe quel navigateur graphique
- Fonction Aucun style de Firefox
- Extension Firebug de Firefox
- Valideur CSS : <http://jigsaw.w3.org/css-validator/>

Références

- Les spécifications de CSS :
 - ▶ <http://www.w3.org/TR/REC-CSS1>
 - ▶ <http://www.w3.org/TR/CSS21/>
 - ▶ <http://www.w3.org/Style/CSS/current-work>
- Wiki de support des standards du Web :
<http://www.webdevout.net/>
- Tests Acid de conformité des navigateurs :
 - ▶ <http://acid1.acidtests.org/>
 - ▶ <http://acid2.acidtests.org/>
 - ▶ <http://acid3.acidtests.org/>
- Des livres :
 - ▶ *HTML et XHTML : La Référence*, O'Reilly
 - ▶ *CSS : La Référence*, O'Reilly

Plan du cours

- 1 CSS et HTML
- 2 Sélecteurs de CSS
- 3 Mise en forme
- 4 Mise en page
- 5 Fonctionnalités mal supportées
- 6 Outils et Références
- 7 Application**

Reproduire la mise en forme de la page modèle n° 2 sur le site du cours.

Règles du jeu

- Interdiction de modifier le HTML.
- La feuille de style CSS doit être valide !