

Conception Web : introduction   CSS

Sarra El Ayari

30 janvier 2008



Plan du cours

- 1 CSS k zakos ?
- 2 Le langage
- 3 Ne pas oublier de valider sa page Web !

CSS kézako ?

- CSS = *Cascading StyleSheets* ou *feuilles de style en cascade*
- On parle de **pollution** du langage HTML à cause de son manque de structuration et clarté.
- C'est le **W3C** (*World Wide Web Consortium*) qui a créé la recommandation CSS en 1996.
- CSS permet d'améliorer l'apparence des documents en lui enlevant toute instruction sur le style des éléments.
- Les éléments de mise en forme sont centralisés et stockés dans un fichier à part : une **feuille de style CSS**.

Histoire de CSS

- Au début langage structurel,
- Puis orientation vers un langage de présentation (italique, gras, souligné, ...).
- Nombres de balises présentes par rapport à l'information réellement utile est stupéfiant !
- Exemple : utilisation balise **font** pour un titre de page :
 - Aucune signification structurelle !

Un manque de structuration

- Les pages non structurées gênent l'accès au contenu informationnel pour les moteurs de recherche (indexation sur titres de pages, entêtes, paragraphes).
- Cela réduit également l'**accessibilité** : navigateurs vocaux ne s'y retrouvent pas.
- Code pas clair, illisible, peu partageable.

En résumé

Véritable **pollution** du code HTML !

Le W3C

- Solution créée par le W3C (et Tim Berners-Lee) en 1996.
- CSS permet l'amélioration de l'apparence des documents, sans toucher à la structuration de la page.

Consignes

- CSS n'aime pas les tableaux ...
- La page est désormais structurée par des balises **<div>** !
- Il s'agit de conteneurs, de boîtes ou encore de blocs dans lesquels ranger vos éléments.
- Ces balises vont vous permettre de définir les parties de votre page et d'y appliquer des styles.

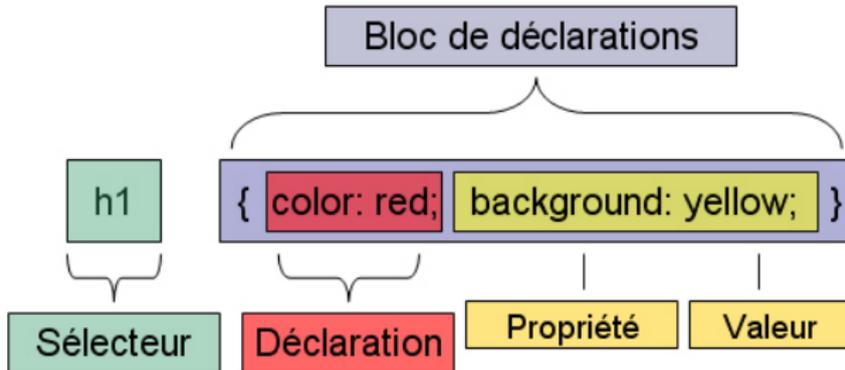
Utiliser une feuille de style

- Pour lier un document XHTML à une feuille de style, il faut insérer la ligne suivante dans la balise **head** :

Syntaxe à employer

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

Structure d'une règle



Rappel

Attention à bien **respecter la syntaxe** pour que votre feuille de style soit correcte !!

Intérêt de CSS

- **Une seule règle** pour spécifier l'apparence de tous les éléments de même type (h1, p, li, ...).
- **Séparation forme et fond** avec un document XHTML qui ne contient que du texte, et aucun élément de mise en forme.

Exemple

Document XHTML :

```
<body>

  <h3> Mise en forme de mon document </h3>

  <p> Plan </p>

  <ul>
    <li> 1er élément </li>
    <li> 2e élément </li>
    <li> 3e élément </li>
  </ul>
</body>
```



Feuille de style :

```
body{ /* Style à appliquer sur l'ensemble du document */
  background-color: #CDC7E1; /* couleur de fond */
  background-image: url(book.jpg); /* image */
  background-repeat: no-repeat; /* non répétée */
  background-position: right; /* et positionnée à droite */
}

h3{ /* Style à appliquer sur les balises de niveau h3 */
  text-align: center; /* texte centré */
}

p{ /* Style à appliquer sur les paragraphes */
  color: red !important; /* texte en rouge */
}

ul{ /* Style des listes à puces*/
  list-style-type: square; /* puces carrées */
}

li{ /* Contenu des listes à puces */
  color: blue; /* police bleue */
  font-style: italic; /* et penchée */
  text-decoration: underline; /* et surlignée */
}
```

Les cascades

- Une cascade est une combinaison de plusieurs styles qui proviennent de sources différentes et qui sont appliquées à un même document, en fonction de priorités.
- Comment résoudre deux instructions qui portent sur le même type de balises ?
 - `h1 color : blue ;`
 - `h1 color : green ;`

Il y a des règles de cascades :

- Trouver toutes les déclarations qui contiennent un sélecteur correspondant à un élément donné.
- Trier les déclarations selon leur poids (**!important**, style de l'auteur)
- Plus une déclaration apparaît tardivement, plus elle aura de poids.

Classes

- On peut utiliser l'attribut **class** sur n'importe quelle balise dans le document XHTML.

Exemple

Dans le document XHTML :

```
<p class="introduction"> Pour commencer, nous allons ...</p>
```

Dans la feuille de style :

```
.introduction {  
color : red ;  
}
```

Selecteurs simples, multiples, universels

- Un **selecteur simple** porte sur le nom d'une balise.
- Un **selecteur multiple** porte sur plusieurs sélecteurs simples, séparés par une virgule.
- Un **selecteur universel** : `*`, qui sélectionne tout.

Exemples

- `body { background-color : pink ;}` indique la couleur du fond d'écran.
- `h1, h2, h3, h4, h5, h6 { color : red ;}` met l'ensemble des titres en rouge.
- `* { color : blue ;}` indique que tous les éléments seront de police bleue.

Sélecteurs de classes

- Fonctionne avec l'attribut *class* apposé à certaines balises du document XHTML.
- Vous noterez le nom de la classe préfixée d'un point . .
- Syntaxe : **élément.NomClasse**

Exemples

- `p.personne {color : pink ;}` met l'ensemble des balises de classe `personne` en rose.

Sélecteurs d'identifiants

- L'identifiant est défini par l'attribut **id** d'une balise XHTML.
- Le sélecteur d'identifiant correspond au nom, préfixe d'un **dièse**.
- Syntaxe : **élément#nomIdentifiant**

Exemples

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

Sélecteurs contextuels

- Il s'agit d'au moins 2 sélecteurs séparés par des espaces. A B sélectionne uniquement les B qui sont contenus dans A.

Exemples

- `h1 em { color : blue ; }` met en bleu les mots en emphase contenus dans un titre h1.

Pseudo-classes

- Les pseudo-classes permettent de sélectionner un élément, uniquement dans certains contextes.

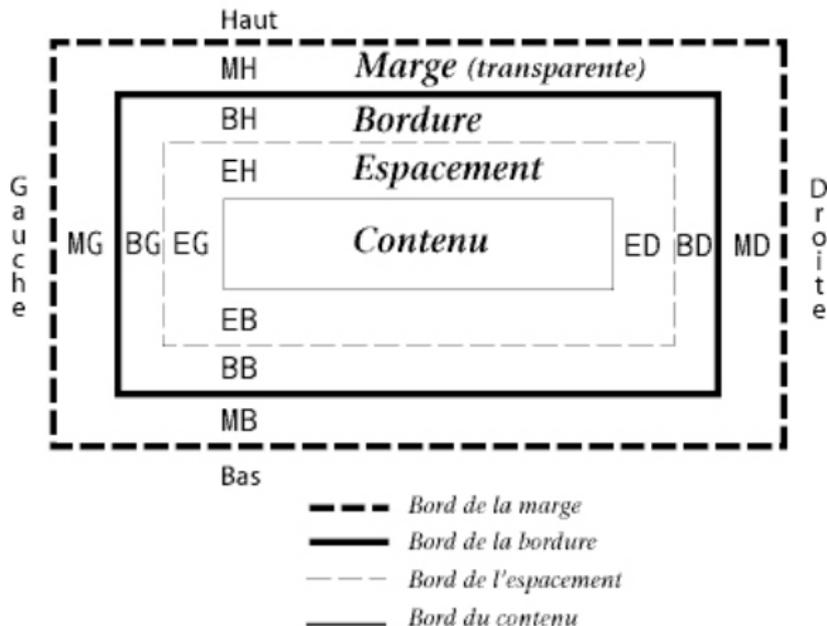
Exemples pour les liens

- **a :link** pour les liens hypertextes.
- **a :visited** pour les liens déjà visités.
- **a :hover** pour le comportement des liens quand on les pointe.
- **a :active** pour le comportement des liens que l'on clique.
- Pour qu'ils ne soient pas soulignés :
text-decoration : none ;

Les boîtes

- Les éléments de type bloc produisent une boîte par défaut qui ne peut avoir aucun élément à ses côtés.
- Ces éléments génèrent une "rupture" avant et après la boîte d'élément.
- Il devient possible de créer une boîte pour chaque objet (ex. entête, menu, pied de page, etc.).

Schéma d'encadrement



Gestion des blocs

- Structurer le document XHTML avec des blocs : ... </div>
- **width** : attribut pour définir la largeur du bloc (**auto** par défaut).
La largeur d'un element est définie comme la distance entre le bord interne gauche et le bord interne droit.
- **height** : attribut pour définir la hauteur des blocs (**auto** par défaut).
La hauteur est la distance entre le bord interne haut et le bord interne bas.

Marges

- Une marge permet de créer un espace blanc autour d'un élément.
- On pourra définir la marge en haut (**margin-top**), à droite (**margin-right**), à gauche (**margin-left**) et en bas (**margin-bottom**) d'un bloc.

Espacement

- Il s'agit de la zone comprise entre les bordures et la zone de contenu.
- Comme pour les marges, il existe 4 attributs : **padding-top**, **padding-right**, **padding-left** et **padding-bottom**.

Flottement

- Le flottement est d fini par la propri t  **float**, qui peut prendre les valeurs :
 - left :  l ment contourne par la droite
 - right :  l ment contourne par la gauche
 - none : valeur par d faut

Positionnement

- Le positionnement est précisé par la propriété **position**, pour laquelle il existe 4 types de positionnement :
 - relative : permet de décaler un élément à une certaine distance
 - absolute : position par rapport au conteneur
 - fixed : position par rapport a la fenêtre d'affichage (ne fonctionne pas avec IE)
 - static : comportement par défaut

Le texte

- Couleur de la police : **color : red ;**
- Police et taille : **font : 1em "Comic Sans ms" ;**
- Positionnement : **text-align : left ;**
- En gras : **font-weight : bold ;**
- En italique : **font-style : italic ;**

Le background

- Pour insérer une image :
background-image : url(image.jpg) ;
- Pour ne pas la répéter : **background-repeat : no-repeat ;**
- La position : **background-position : left top ;**

Les bordures

- Pour créer une bordure : **border : 1px solid blue**

Elle peut également prendre la valeur **dashed** (tirets), **dotted** (pointillés), **groove** (en relief).

Les listes à puces

Il est possible d'intervenir sur les puces elles-mêmes :

- Pas de puces : **liste-style : none ;**
- Des puces carrées : **liste-style : square ;**

Pour spécifier la manière dont un élément va s'afficher : **display**

- **display : block** ; affichera les éléments sous la forme de boîtes, mises les unes en dessous des autres (cf. listes à puces)

Il existe d'autres attributs possibles, que nous ne verrons pas ici.

Commentaires

Syntaxe

Pour insérer des commentaires dans une feuille de style :

```
/* votre commentaire */
```

Valider ses documents

- XHTML
 - En ligne sur <http://validator.w3.org/>
- CSS
 - En ligne sur <http://jigsaw.w3.org/css-validator/>

Déroulement du TP

- Sur la page du cours [▶ par ici](#)
 - N'oubliez pas d'**indenter** votre code ...
 - ... et d'insérer des **commentaires** !

Pour approfondir

- En ligne :
 - A voir absolument : [Le site de Zen Garden !](#)
 - Liste des options à utiliser : [Sur le site du Zéro.](#)
 - La définition proposée par [Wikipedia.](#)
- Sur papier :
 - **CSS - La référence**, Eric A. Meyer, 2005, éd. O'Reilly
 - Un petit livre récapitulatif des fonctions CSS :
CSS, précis et concis, Eric A. Meyer, 2004, éd. O'Reilly

► Des remarques ?