

Création et Manipulation de documents

(Hélène Renard / Sylvain Schmitz)

Travaux Dirigés – Séance n°6

1 Objectifs du TD

Utiliser les feuilles de style pour mettre en forme une page XHTML.

2 Introduction

L'abréviation CSS signifie *Cascading Style Sheets*, soit en français : « feuilles de style en cascade ».

CSS est un langage de style qui définit la présentation de documents XML, et en particulier des documents XHTML. Par exemple, CSS couvre les polices, les couleurs, les marges, les lignes, la hauteur, la largeur, les images d'arrière-plan, les positionnements évolués et bien d'autres choses. En ce sens, les feuilles de style sont complémentaires d'une page XML, celle-ci décrivant ce qui doit être affiché (structure le contenu) et les feuilles de style décrivant comment l'afficher (formate un contenu structuré – la forme) (cf. figure 1).

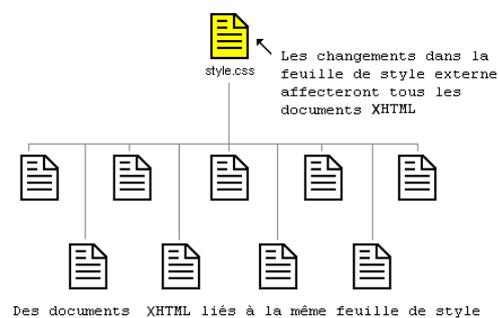


FIG. 1 – Une seule CSS pour de nombreux fichiers XHTML

Vous allez découvrir quelques possibilités de mise en forme de CSS dans les sections qui suivent. La version officielle de la norme actuelle pour les feuilles CSS est actuellement la 2.1 <http://www.w3.org/TR/CSS21/>, tandis qu'une version 3 est assez avancée pour avoir droit à un support partiel de la part de plusieurs navigateurs Internet.

3 Fonctionnement d'une CSS

Lorsque votre navigateur Web préféré affiche un document, il doit combiner le contenu du document avec ses informations de style. Ainsi, il traite le document en deux étapes :

1. Il convertit le langage de balisage et la CSS dans un DOM (*modèle objet de document*). Le DOM représente le document dans la mémoire de l'ordinateur. Il combine le contenu du document et son style.
2. Il affiche le DOM.

DOM est essentiellement utilisé pour pouvoir modifier facilement des documents XML. Il a une structure sous forme d'arbre. Chaque balise et morceau de texte figurant dans le langage de balisage (XHTML pour le TD) devient un nœud dans la structure arborescente (cf. figure 3).

```
<p>
  <strong>C</strong>ascading
  <strong>S</strong>tyl
  <strong>S</strong>heets
</p>
```

FIG. 2 – Code XHTML

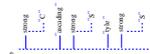


FIG. 3 – DOM correspondant au code XHTML de la figure 2

La compréhension du DOM aide à créer, déboguer et maintenir votre CSS, car le DOM est l'endroit où CSS et le document se rencontrent.

Passons maintenant au détail de la CSS elle-même.

Une CSS comprend une ou plusieurs règles décrivant comment les éléments d'une page correspondant à des éléments XML spécifiques devraient être affichés. Chaque règle d'une feuille de style en cascade comprend deux parties :

1. le sélecteur (XHTML, classe ou `id`) : un caractère alphanumérique qui identifie une règle ; le plus souvent un élément simple tel que `body` ou `h1`
2. la déclaration qui se compose
 - (a) d'une propriété : elle identifie ce qu'il faut définir
 - (b) d'une valeur de cette propriété (ce peut être un mot-clé comme `oui` ou `non`, un pourcentage, un nombre. dépend uniquement de la propriété à laquelle cette valeur est assignée)

Exemple de règles d'une CSS :

```
h1 {
  font-size: 12px;
}
```

Dans cet exemple `h1` est le sélecteur, `font-size` est la propriété et la valeur est `12px`. Notez les deux points après le nom de la propriété et le point virgule entre chaque déclaration. Chaque élément possède ses propres propriétés (par exemple, la propriété `list-style` pour l'affichage des listes) mais quelques propriétés peuvent être utilisées avec tous les éléments.

La syntaxe de base d'une règle CSS est donc de la forme :

```
selecteur_1, selecteur_2, ..., selecteur_m {
  propriété_1: valeur_1;
  propriété_2: valeur_2;
  . . .
  propriété_n: valeur_n;
}
```

4 Les différentes propriétés utilisables dans une CSS

4.1 Couleur et arrière-plan d'un élément

background-color : permet de spécifier une couleur pour l'arrière plan. Les valeurs peuvent être soit un nom de couleur (mais il n'en existe qu'une dizaine de prédéfinies) soit une

valeur hexadécimale de la forme #xxxxxx où x est un chiffre hexadécimal (le raccourcis #xyz désigne la couleur #xxyyzz). Plusieurs moyens s'offrent à vous pour connaître la valeur hexadécimale d'une couleur. Par exemple, vous pouvez utiliser un logiciel de graphisme (comme gimp) ou trouver une ressource sur le web (comme la page http://www.w3schools.com/css/css_colornames.asp).

background-image : permet de placer une image en arrière plan pour un élément sélectionné. Il suffit d'indiquer l'emplacement de l'image voulue, comme par exemple :

```
div.funny {
  background-image: url("http://www.acme.com/images/funnyface.gif");
}
```

background-repeat : permet de répéter l'image d'arrière plan, et cela de manière horizontale, verticale, horizontale et verticale ou sans répétition.

background-position : permet de positionner votre arrière-plan où vous le souhaitez. Par défaut, il sera en haut à gauche de l'écran. Il y existe plusieurs méthodes pour fixer la valeur de cette propriété. Par contre, elles se présentent toutes sous la forme d'un jeu de coordonnées. Par exemple, la valeur « 100px 200px » positionne l'image d'arrière-plan à 100px depuis la gauche et à 200px depuis le haut de la fenêtre du navigateur. Les coordonnées peuvent s'exprimer en pourcentages de la largeur de l'écran, ou en unités fixes (pixels, centimètres, etc.), ou on peut utiliser les mots-clés **top**, **bottom**, **center**, **left** ou **right**.

background : est un raccourci pour toutes les propriétés listées précédemment. Si une propriété manque, elle prend automatiquement sa valeur par défaut. Vous pouvez, par exemple écrire :

```
body {
  background: url("../art/hwslogo.gif") white repeat-x;
}
```

color : permet de définir une couleur pour le texte de l'élément sélectionné. Les valeurs permises sont les mêmes que pour la propriété **background-color**.

4.2 Police d'un élément

font-family : permet d'indiquer la liste prioritaire des polices à utiliser pour l'affichage d'un élément donné ou d'une page Web. Si la première police de la liste n'est pas disponible sur l'ordinateur utilisé pour accéder au site, la police suivante est essayée et ainsi de suite jusqu'à ce qu'il y en ait une qui convienne. Dans l'exemple suivant, les titres marqués par des éléments **<h1>** s'afficheront dans la police **Arial**. Si elle n'est pas installée dans l'ordinateur de l'utilisateur, la police **Verdana** sera utilisée à la place. Si ces deux polices sont indisponibles, les titres s'afficheront dans une police de la famille **sans-serif**.

```
h1 {
  font-family: arial, verdana, sans-serif;
}
```

font-style : pour spécifier le style de police. Les valeurs possibles sont **normal**, **italic** ou **oblique**.

font-variant : permet de choisir entre les variantes normal ou small-caps d'une police (*une police small-caps utilise des lettres en capitales de taille réduite à la place des lettres en minuscules.*)

font-weight : décrit avec quel degré de « noir » présenter la police. Elle peut avoir un degré de « noir » normal ou **bold** (gras). Certains navigateurs permettent même d'utiliser des nombres entre 100 et 900 (les centaines) pour décrire ce degré.

font-size : comme son nom l'indique, permet de régler le corps d'une police. Les différentes unités utilisées sont pixels, pourcentages, points etc.

font : est un raccourci pour toutes les propriétés listées précédemment. Vous pouvez par exemple écrire :

```
p {
  font: italic bold 30px arial, sans-serif;
}
```

4.3 Texte d'un élément

text-indent : permet d'ajouter une touche d'élégance aux paragraphes de texte en appliquant une indentation à la première ligne du paragraphe.

text-align : permet de définir l'alignement horizontal d'un texte. Les valeurs possibles sont `left`, `right`, `center` ou `justify` (pour que les marges gauches et droites soient régulières).

text-decoration : permet de souligner le texte, le barrer, ou tracer un trait au-dessus, etc. Les valeurs possibles sont `underline`, `overline`, `line-through`.

letter-spacing : permet de définir l'espacement entre les caractères du texte.

Exercice n°1 :

1. Créez un fichier `style.css` dans votre répertoire `CMDocs/CSS` et définissez une nouvelle couleur de fond (par exemple la couleur `#c8c8c8`) pour la page correspondant au fichier `~hrenard/tp_css.xhtml`. (Copiez cette page XHTML dans votre répertoire `CMDocs/CSS`.)
2. Rajoutez la ligne suivante dans l'entête de la page :
`<link href="style.css" rel="stylesheet" type="text/css" />`.
Vérifiez que la couleur de fond est bien appliquée.
3. À l'aide d'une seule définition, spécifiez une même couleur pour les éléments `h1` et `h2`.
4. Le fait de définir des propriétés pour plusieurs éléments en même temps n'empêche pas de rajouter plus loin dans la feuille de style des définitions pour un élément particulier de la liste. Rajoutez une définition pour que l'élément `h1` soit de plus centré sur la page.
5. De même qu'il existe un vérificateur pour les pages XHTML, il existe aussi un vérificateur pour les feuilles de style. Rendez vous sur la page <http://jigsaw.w3.org/css-validator/> pour valider votre feuille ou la corriger s'il y a des erreurs.

5 Les sélecteurs simples

5.1 Les sélecteurs de classe

Il est parfois utile de pouvoir définir plusieurs comportements différents pour un même élément. Pour cela, on utilise des sélecteurs de classe qui sont de la forme `element.nom_de_classe` (le corps de la définition suit les mêmes règles qu'avant). Pour que cette définition s'applique dans le document XHTML, il faut alors rajouter à l'élément `element` l'attribut `class="nom_de_classe"`.

Exemple de sélecteur de classe :

```
h1.alerte {
  color: red;
}
```

Exemple de code XHTML associé au sélecteur de classe :

```
<h1 class="alerte">Danger de Mort !</h1>
<p class="alerte">Soyez Prudent...</p>
```

Une autre façon d'utiliser les sélecteurs de classe est la suivante :

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

`alerte` est ici dans ce cas une classe indépendante que vous pouvez employer avec n'importe quelle balise XHTML. Une classe peut être directement associée à un sélecteur XHTML auquel cas on parle de classe dépendante. Autrement dit, le sélecteur de classe ne peut être employé qu'avec une balise XHTML particulière.

Exercice n°2 :

Définissez deux comportements différents pour l'élément `p` : l'un général qui justifie les lignes d'un paragraphe, et un autre spécial qui affiche le texte en rouge. Éditez le document XHTML pour que l'avant dernier paragraphe applique cette dernière propriété.

Exercice n°3 :

On va changer quelque peu l'affichage du tableau :

1. Définissez une nouvelle couleur de fond pour le tableau et décalez-le vers la droite.
2. Définissez une autre couleur de fond pour l'affichage des noms des colonnes.
3. En général, l'affichage des titres de livres se fait à l'aide d'une police italique. Réalisez cet affichage.

5.2 Les sélecteurs id

Le sélecteur `id` a presque la même fonction, à la différence importante qu'on ne peut l'utiliser qu'une seule fois dans la page, contrairement au sélecteur `class`. C'est pour cela qu'il est plutôt utilisé à la mise en page qu'à la mise en forme de caractères.

La syntaxe générale pour affecter un identifiant unique à un élément XHTML particulier, il suffit de spécifier la valeur de l'identifiant et la faire précéder d'un signe dièse `#`.

Exemple de sélecteur id :

```
#vert {  
  color: green;  
}
```

Exemple de code XHTML associé au sélecteur id :

```
<span id="vert">Un joli texte vert</span>
```

6 Les sélecteurs avancés

6.1 Les sélecteurs descendants

Un sélecteur descendant est constitué de deux sélecteurs simples ou plus séparés par un blanc. Il cible les éléments qui sont des descendants d'un élément correspondant au premier sélecteur simple.

```
td em {  
  instructions...  
}
```

L'exemple précédent applique un style à un bloc `em` contenu dans une cellule `td` de table, que ce bloc soit un descendant direct ou non : il s'appliquerait ainsi à `<td><q>Avez-vous lu Le procès?</q></td>`. Les sélecteurs descendants permettent aussi de cibler des éléments sans pour autant leur donner une classe ou un `id`, ce qui donne la possibilité de garder un code simple et clair.

6.2 Les sélecteurs d'enfants

Un sélecteur d'enfant cible un descendant immédiat d'un élément donné. Ce sélecteur se compose de deux sélecteurs ou plus séparés par un signe « supérieur à » (« > »). Le parent est à gauche du combinateur, et des espaces sont autorisés autour de ce dernier.

```
td > ul {
  instructions...
}
```

L'exemple précédent applique un style à une liste `ul` contenue dans une cellule `td` de table. L'élément modifié doit être un enfant direct du premier élément.

6.3 Les sélecteurs adjacents

Un sélecteur adjacent est construit par deux sélecteurs simples séparés par un signe « plus » (« + »). Des espaces sont autorisés autour du combinateur. Ce sélecteur cible le second élément qui suit immédiatement le premier. Autrement dit, ces éléments doivent avoir le même parent et le premier élément (à gauche du combinateur) doit précéder immédiatement le second :

```
p + p {
  color:#F00;
}
```

La règle ci-dessus appliquée à l'exemple suivant n'affectera que « Paragraphe 2 » :

```
<div>
  <p>Paragraphe 1</p>
  <p>Paragraphe 2</p>
</div>
```

6.4 Les sélecteurs de pseudo-classes

6.4.1 Pseudo-classe d'ancres et dynamiques

La dernière forme de sélecteur que nous verrons dans ce TD concerne celle des pseudo-classes, de la forme `element:pseudo_classe`. Elle est surtout employée pour changer l'affichage des liens dans une page web : `a:link` pour les liens non visités, `a:visited` si l'internaute a déjà cliqué sur ce lien, `a:hover` lorsque la souris passe au dessus du lien et `a:active` lorsque l'internaute est en train de cliquer sur le lien.

Exemple de sélecteur de pseudo-classes :

```
a:link { color: red; } /* lien non-visit   */
a:visited { color: blue; } /* lien visit   */
a:hover { color: yellow; } /* lien survol   */
a:active { color: lime; } /* lien activ   */
```

Exercice n  4 :

1. Définissez une couleur différente pour chacun des événements ci-dessus et testez le résultat.
2. Validez votre feuille ou corrigez-la s'il y a des erreurs.

6.4.2 Pseudo-classes `first-child`

La classe `:first-child` s'applique à un élément étant le premier enfant d'un élément parent. L'exemple suivant fait en sorte que le premier `<p>` inclus immédiatement après l'ouverture d'un `<div>` reçoit une marge à gauche de 15 pixels.

```
div:first-child p {  
  margin-left: 15px;  
}
```

6.5 Les sélecteurs de pseudo-éléments

Les pseudo-éléments ont le caractère deux-points en début de leur nom. Ils désignent des éléments qui ne correspondent pas directement à une balise comme par exemple `:first-letter` que vous avez vu dans les sous-sections suivantes.

6.5.1 Pseudo-éléments `first-line`

Le pseudo-élément `first-line` applique un style particulier à la première ligne d'un élément.

```
p:first-line {  
  color: blue;  
  font-variant: small-caps;  
}
```

6.5.2 Pseudo-éléments `first-letter`

Le pseudo-élément `first-letter` applique un style particulier à la première lettre d'un élément. Cela vous permet par exemple de faire des jolis débuts de paragraphe avec une lettrine.

```
p:first-letter {  
  font-size: 200%;  
  font-style: italic;  
  color: green;  
}
```