

## TD 2 : CSS

7 octobre 2015

**Objectif:** Ce TD a pour but d'étudier la mise en forme de code HTML grâce aux feuilles de style CSS (**C**ascading **S**tyle **S**heets). À l'issue du TD vous devez savoir

- créer une feuille de style CSS ;
- comprendre la différence entre élément et classe d'éléments HTML ;
- définir plusieurs classes pour un même élément ;
- définir des sous-classes qui héritent des propriétés de leur parents ;
- gérer les éléments comme des boîtes ;
- modifier l'arrière-plan d'un élément HTML.

### 1 Une question de style

Une feuille de style, comme son nom l'indique, sert à définir un style. Mais qu'est ce qu'un style en HTML? Un style est un ensemble de règles permettant de définir comment un objet doit être rendu (i.e. dessiné, affiché).

Une règle CSS est définie par un *sélecteur*, l'objet auquel le style va s'appliquer, et un *bloc de déclaration*, contenant la liste des attributs (la syntaxe est `propriété : valeur ;`).

Par exemple, le code suivant :

```
p {  
    color: red;  
    text-align: center;  
}
```

indique que le texte de tous les paragraphes (correspondant à la balise `<p>`) sera en rouge et centré par rapport à l'élément parent.

Où peut-on définir un style et quel est l'avantage par rapport aux attributs vus dans le TD précédent? Il y a trois manières de définir un style :

- *intégré* directement dans les balises HTML ;
- avec une feuille de style *interne* ;
- avec une feuille de style *externe*.

La première méthode est utilisée pour appliquer un style à un seul élément. La syntaxe est similaire à celle vue dans le TD précédent.

```
<h1 style="color:blue; margin-left:30px;
text-align:center;">Un beau titre bleu, centré
avec une marge à gauche de 30 pixels.</h1>
```

Afin d'avoir tous les titres de niveau 1 de la page avec le même style, il est préférable d'utiliser la deuxième méthode et de définir les instructions au début du document, notamment dans l'en-tête du document. Pour cela, on utilisera les balises `<style>` et `</style>`.

```
<head>
  <style>
    h1 {
      color: orange;
      margin-left: 15px;
      text-align: right;
    }
  </style>
</head>
```

La troisième méthode consiste à créer un fichier `.css` contenant toutes les définitions de style et de l'inclure dans l'en-tête de notre page HTML. Par exemple, le fichier `mon_style.css`:

```
h1 {
  color: green;
  text-align: center;
}

h3 {
  color: red;
  text-align: justify;
}
```

peut être utilisé dans une page HTML avec le code :

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

Cette méthode permet de partager un même style entre différentes pages, voir différents site web. On utilisera cette méthode durant tous les TDs. De cette façon, on pourra utiliser les mêmes styles dans plusieurs pages, et il sera plus facile de changer une propriété sans devoir modifier chaque fichier `.html`.

Dans le cas où plusieurs styles ont été spécifiés pour le même élément HTML, la priorité sera donnée au style intégré dans la balise, suivi par les indications de la feuille de style (interne et externe) et enfin, les paramètres du navigateur.

► **Exercice 1 :** À l'aide des exemples ci-dessus, créez une feuille de style `my_way.css` permettant de centrer les titres de niveau 1 et de les colorer en rouge; aligner à droite les titres de niveau 2; justifier les paragraphes avec une marge à gauche de 10 pixels.

Ensuite, créez une page HTML appelant dans l'en-tête la feuille de style que vous venez de créer et contenant dans le corps le code suivant :

```
<h1> Comme d'habitude </h1>

<h3> écrite par Claude Fran&ccedil;ois</h3>

<p>Je me lève et je te bouscule<br>
Tu n'te réveilles pas<br>
Comme d'habitude.</p>
```

Ouvrez maintenant la page HTML dans un navigateur. Que remarquez-vous?

## 2 Quel beau paysage

Vous voyez la vie en rose, même si vous êtes en rouge et que vous n'avez pas le sang bleu? Vous vous sentez un merle blanc parce que vous êtes un vrai cordon bleu, avez la main verte et tout le monde vous donne carte blanche? Dans ce cas-là, c'est l'heure d'ajouter un peu de couleur dans votre vie. En partant de l'arrière-plan de votre page web.

Pour ce faire, on modifiera le style de l'élément `body` en utilisant le code suivant :

```
body {
  background-color: #00FF00;
}
```

En CSS, une couleur est définie par :

- un valeur hexadécimal, comme par exemple "#00FF00";
- un valeur RGB (rouge-vert-bleue), comme par exemple "rgb(0,255,0)";
- un nom de couleur (en anglais), comme par exemple "green".

Dans l'exemple suivant on définit différentes couleurs pour les éléments `h1`, `p` et `table` :

```
h1 {
  background-color: red;
}

p {
  background-color: #F7C942;
}

table {
  background-color: rgb(AA,00,AA);
}
```

Avec `background-image` on peut utiliser une image comme arrière-plan d'un élément. Par défaut, l'image est répétée (à la fois horizontalement et verticalement) de manière à couvrir entièrement l'élément. Pour modifier cette propriété on utilisera `background-repeat` en choisissant une des valeurs :

- `repeat-x`, pour que l'image soit répétée que horizontalement ;
- `repeat-y`, pour que l'image soit répétée que verticalement ;
- `no-repeat`, pour que l'image ne soit pas répétée.

Par exemple, le code :

```
body {
  background-image: url("images/tulipes.jpg");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: right top;
}
```

permet d'afficher une jolie image avec des tulipes dans le coin en haut à droite de la page. Ajoutez un très long texte (en utilisant, par exemple, le site <http://fr.lipsum.com>) et faites défiler la page. Que remarquez-vous?

Modifiez maintenant la valeur de la propriété `background-attachment` par `scroll`. Quel est le changement?

► **Exercice 2** : Modifiez la feuille de style `my_way.css` afin d'ajouter une image d'arrière-plan qui remplit toute la page et qui surligne le texte de tous les paragraphes en jaune. Testez ce nouveau style dans la page `recherche.html` que vous avez créée la semaine dernière.

### 3 En quelle classe voulez-vous voyager ?

Il arrive assez souvent qu'on ait besoin de plusieurs styles pour un même élément. Pour ce faire, on définira des *classes*, c'est-à-dire des groupes de propriétés pour un (ou plusieurs) élément(s).

Supposons, par exemple, qu'on veuille définir les styles suivants pour les paragraphes :

- *classique*, où le texte est noir et justifié ;
- *stabiloté*, lui aussi en noir et justifié, mais aussi en italique et avec le fond coloré en jaune.

Pour cela, on écrira dans notre feuille de style :

```
p.classique {
  color: black;
  text-align: justify;
}

p.stabilote {
  color: black;
  text-align: justify;
  font-style: italic;
  background-color: yellow;
}
```

tandis que dans notre page HTML on utilisera le code :

```
<p class="classique">Ceci est un texte justifié, en
noir... bref : quelque chose de très ennuyeux !</p>

<p class="stabilote">Tandis que cette phrase contient
des informations très importants et il vaut mieux de
le stabiloter !</p>
```

Si maintenant on essaie d'utiliser le style `stabilote` sur un autre type d'élément, par exemple :

```
<h2 class="stabilote">Un titre important.</h1>
```

on va être un peu déçu.

Et pourtant les propriétés `color`, `text-align`, `font-style` et `background-color` ont du sens à la fois pour les éléments du type `p` et pour les `h2`.

Pour résoudre ce problème on pourrait définir le même style pour les deux :

```
p.stabilote, h2.stabilote {
  color: black;
  text-align: justify;
  font-style: italic;
  background-color: yellow;
}
```

mais, dans ce cas-ci, on aura le même problème avec, par exemple, les éléments du type `h3`.

Si on veut définir une classe qui peut s'appliquer à n'importe quel type d'élément, il suffit de laisser vide la partie à gauche du point. Par exemple, la classe `bien_visible` défini par :

```
.bien_visible {
  color: red;
  text-align: center;
}
```

permet de centrer et de mettre en rouge le contenu de tous les éléments qui ont “`bien_visible`” comme classe.

► **Exercice 3 :** En utilisant les exemples précédents, créez les classes suivantes (aidez-vous de la page [http://www.w3schools.com/css/css\\_text.asp](http://www.w3schools.com/css/css_text.asp)):

- *classique*, le texte est noir et justifié ;
- *stabilote*, comme le classique mais avec le fond coloré en jaune ;
- *erreur*, le texte est barré et en bleu ;
- *citation*, le texte est en italique, aligné à gauche et avec une marge à gauche de 10 pixels ;
- *excentrique*, le texte est aligné à droite, en gras, en italique, souligné et avec la première lettre de chaque mot en majuscule.

On devra pouvoir utiliser ces classes pour n'importe quel type d'élément.

Parfois, il peut être utile de définir des propriétés s'appliquant à un unique élément qui ne peut apparaître qu'une seule fois dans la page.

Pour ce faire, on utilisera des *id*. La syntaxe du sélecteur `id` est très similaire à celle du sélecteur `class`. Dans la feuille de style on remplacera le “.” par un “#”.

Par exemple :

```
#grand_titre {
  color: red;
  text-align: center;
  font-size: 40px;
}
```

Tandis que dans le code HTML on écrira:

```
<h1 id="grand_titre">Ceci est LE titre de la page</h1>
```

► **Exercice 4 :** Définissez dans votre feuille de style un id `premier` qui permet d'aligner le texte d'un élément à droite. Testez maintenant le code HTML suivant :

```
<p id="premier">Ceci est un paragraphe.</p>
<p>Ceci est un autre paragraphe.</p>
```

## 4 La boîte de Pandore

Si on veut pouvoir tirer le maximum possible du couple HTML/CSS, il est important de comprendre comment le navigateur interprète les balises lors du rendu du document à l'écran. Pour cela, il faut connaître la notion de "boîte".

Pour expliquer les choses simplement, chaque élément peut être vu comme une boîte. Chaque boîte, en plus de son contenu, possède 3 bords imbriqués les uns dans les autres: une marge interne (*padding*), une bordure (*border*) et une marge externe (*margin*).

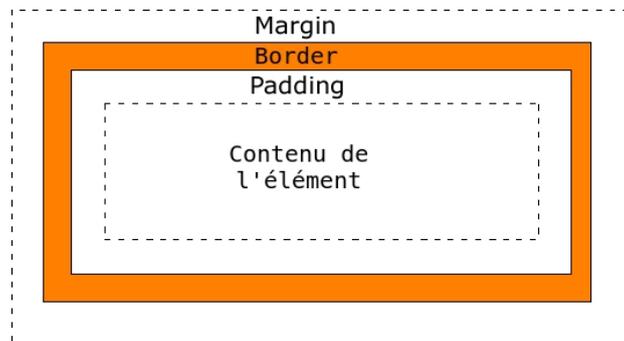


FIG. 1 – Représentation d'un élément HTML sous forme de boîte.

La Figure 1 montre à quoi correspondent ces bords. Remarquez que les marges interne (*padding*) et externe (*margin*) sont transparents. Les dimensions du contenu de la boîte sont contenues dans les attributs `width` et `height`.

Un exemple de boîte et de comment on peut définir un style en CSS est donnée ci-dessous (où l'on introduit la balise `<div>`):

```
div {
  width: 500px;
  padding: 30px;
  border: 5px solid #FF9933;
  margin: 10px;
}
```

► **Exercice 5 :** Définissez dans votre feuille de style une classe `chauvin` pour les éléments `div` ayant une marge externe de 14 pixels, une bordure bleue de 7 pixels, une marge interne de 17.89 pixels, l'arrière-plan blanc et le texte en rouge et en italique.

Testez maintenant dans votre page HTML le code :

```

<div class="chauvin">Allons enfants de la Patrie,<br>
le jour de gloire est arrivé !<br>
Contre nous de la tyrannie<br>
l'étendard sanglant est levé.<br>
Contre nous de la tyrannie<br>
l'étendard sanglant est levé.<br>
Entendez-vous dans les campagnes<br>
mugir ces féroces soldats ?<br>
Ils viennent jusque dans vos bras<br>
égorger vos fils, vos compagnes !</div>

```

## 5 Des cascades et des chutes d'eau

Une particularité de CSS est la possibilité de transmettre certaines propriétés par *héritage* (du père au fils) ou par *cascade* (plusieurs spécification de style qui interagissent).

Certains styles, tels que la police de caractère, sa couleur, l'alignement du texte, etc., se transmettent automatiquement des éléments HTML "parents" vers leurs "enfants". D'autres styles tels que la bordure, les marges, etc. ne se transmettent pas.

Considérons par exemple la feuille de style contenant :

```

div {
  color: maroon;
  text-align: center;
  font-family: Times;
  margin-top: 50px;
  padding-right: 15px;
}

```

et le code HTML suivant :

```

<div>
  Cette phrase est écrite en bordeaux, en police Times New
  Roman et est centrée.

  <p>
    Cette phrase aussi est en bordeaux, Times et centrée. Par
    contre, ce paragraphe ne sera pas décalé sur la droite par
    rapport à la première phrase, car la propriété 'padding'
    ne se transmet pas par héritage.
  </p>
</div>

```

Certaines de propriétés de l'élément *parent* `div` seront héritées par son élément *enfant* `p`<sup>1</sup>.

CSS permet aussi de définir des styles qui ne s'appliquent qu'à certains éléments qui descendent d'un élément particulier (ou d'une classe particulière). Considérons, par exemple, le style suivant :

1. En théorie, il est possible de "forcer" l'héritage d'une propriété non-héritable en précisant, par exemple : `<p style="padding:inherit">`. Malheureusement, tous les navigateurs ne supportent pas l'attribut `inherit`.

```
div p
{
    text-indent: 10px;
}
```

Dans cet exemple, tout paragraphe se trouvant à l'intérieur d'un `div` doit voir sa première ligne indentée de 10 pixels. Cette règle n'affecte que les éléments de type `p` descendants d'un élément de type `div`.

On peut également utiliser des classes pour affiner un peu plus la sélection, comme dans les exemples suivants :

```
/* n'affecte que les paragraphes descendants
   d'un élément de type 'div' de classe 'mon_div' */
div.mon_div p
{
    text-indent: 20px;
}

/* n'affecte que les paragraphes de classe
   'mon_paragraphe' et descendants d'un élément de
   type 'div'. */
div p.mon_paragraphe
{
    text-indent: -10px;
}

/* n'affecte que les paragraphes de classe
   'mon_paragraphe' descendants d'un élément de
   type 'div' de classe 'mon_div'. */
div.mon_div p.mon_paragraphe
{
    text-indent: 40px;
}
```

Avez-vous remarqué qu'une partie du code est contenue entre “/\*” et “\*/”? Comment le navigateur traite-t-il ce texte?

► **Exercice 6 :** Créez dans votre feuille de style, une classe `mon_div` pour les éléments `div` telle que l'élément :

- soit placé à 20 pixels à gauche et 40 pixels en haut de l'élément parent ;
- ait une bordure fine et rouge à gauche ;
- ait une bordure épaisse et verte à droite ;
- ait une bordure en pointillés, moyenne et bleue en haut ;
- n'ait pas de bordure en bas ;
- ait un padding de 10 pixels dans toutes les directions ;
- ait une couleur d'arrière-plan #AAAAAA.

► **Exercice 7 :** Créez dans votre feuille de style, une classe `mon_paragraphe` pour les éléments

p tel que l'élément ait :

- une bordure fine et bleue dans les quatre directions ;
- une couleur d'arrière-plan `rgb(255, 0, 255)`.
- son le texte en gras et souligné.

Testez maintenant le code de l'exemple précédent avec celui que vous venez de créer et celui de l'Exercice 6.

Imaginons maintenant que nous voulons appliquer un certain style seulement aux fils d'un certain élément (et pas aux petit-fils ou aux autres descendants).

Pour cela on utilisera la syntaxe “père > fils”, comme par exemple :

```
div > span
{
    background-color: yellow;
}
```

Testez par exemple le style précédent et le code HTML suivant :

```
<div>
  <span>Une phrase dans un div.</span>
  <span>Une autre phrase dans un div.</span>
  <p>
    <span>Une troisième phrase dans un div.</span>
  </p>
</div>
```

## 6 Pseudo mais pas faux

Pour compliquer un peu les choses, on peut utiliser, à côté des classes, aussi les *pseudo-classes*. Une pseudo-classe est un mot-clé ajouté au sélecteur pour indiquer un état particulier de l'élément qui doit être sélectionné.

Les pseudo-classes, ainsi que les pseudo-éléments, permettent d'appliquer un style à des éléments non seulement en relation avec le contenu de l'arborescence du document, mais aussi avec des facteurs externes comme l'historique du navigateur (`:visited` par exemple), l'état du contenu (comme `:checked` pour les formulaires), ou la position du pointeur (comme `:hover` qui permet de savoir si le pointeur survole un élément ou non).

Supposons, par exemple, qu'on veuille colorer en bleue le première paragraphe de chaque div. Pour ce faire, on utilisera la pseudo-classe `first-child`:

```
p:first-child {
    color: blue;
}
```

► **Exercice 8 :**

► **Exercice 9 :** En vous aidant de la page [http://www.w3schools.com/css/css\\_link.asp](http://www.w3schools.com/css/css_link.asp), modifiez le fichier `my_way.css` pour que le texte de liens (et seulement les liens) satisfasse les propriétés suivantes :

- coloré en cyan si le lien n'a pas encore été visité ;

- souligné et coloré en vert lorsque l'utilisateur passe la souris dessus ;
- souligné et coloré en rouge au moment où l'utilisateur clique dessus ;
- coloré en bleu foncé si le lien a déjà été visité.

Créez maintenant une page HTML (ou modifiez une faites dans le TD précédent) avec des liens hypertextes et testez sur celle-ci votre feuille de style.

## 7 À vous de jouer (et de vous intégrer)

Cet exercice est à faire avant la prochaine séance.

► **Exercice 10** : Créez plusieurs petites pages et leur feuille de style mettant en application ce qui a été vu jusqu'à maintenant. Ensuite, publiez les pages sur votre compte étudiant et ajoutez un lien dans la page dédiée sur la plateforme e-learning.

Le sujet de ces pages sera «Le week-end d'intégration à l'IMAC». Vous serez jugés à la fois sur le fond et la forme. Lâchez-vous (et revenez entiers) !