

CSS 3

Julien David

A101 - david@lipn.univ-paris13.fr

2016

Cascading Style Sheets

- **CSS** est un langage de **feuilles de styles**.
- Une feuille de style permet de décrire le rendu d'un fichier HTML.

Attention (encore)

- CSS n'est pas un langage de programmation.

Écriture d'une feuille de style

```
1 selecteur{  
2     propriete 1: valeur;  
3     propriete 2: valeur;  
4 }
```

Écriture d'une règle

- Les sélecteurs les plus simples sont les balises html.
- on verra qu'il existe des sélecteurs plus complexes.

Écriture d'une feuille de style

```
1 html{
2   background-color: grey;
3 }
4
5 body{
6   background-color: blue;
7 }
8
9
10 main{
11   background-color: black;
12   color: white;
13   text-align: center;
14 }
```

Associer une feuille de style à un fichier HTML

Lier CSS et HTML

On ajoute dans l'en-tête du fichier HTML (balise **<head>**) la ligne suivante.

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

Le Formatage du texte

- *font-family* : police de caractère.
- *font-size* : taille des caractères.
- *font-style* : texte en italique ou non.
- *font-weight* : texte en gras ou non.
- *text-align* : alignement horizontal du texte (gauche, droit, centré, justifié).
- *text-decoration* : texte souligné, surligné (trait au dessus), barré, clignotant, normal.
- *color* : couleur du texte.
- ...

Couleurs

Il existe 3 façon de désigner une couleur :

1 par son nom :

red, blue, lightblue, ...

(140 noms supportés à ce jour par tous les navigateurs)

2 par les fonctions *rgb* et *rgba* :

rgb(255, 0, 0), *rgb*(0, 0, 255), *rgb*(173, 216, 230)

3 en hémidécimal :

#FF0000, #0000FF, #ADD8E6

Le *background*

- *background-color*
- *background-image*
- *background-repeat*
- *background-attachment*
- *background-position*

Les bordures

- *border-width*
- *border-style* (requis)
- *border-color*
- *border-radius*

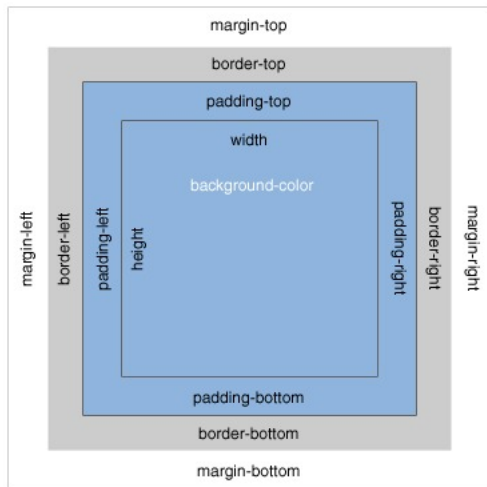
Les listes

- *list-style-type*
- *list-style-image*
- *list-style-position*

Les tableaux

- *border-collapse*

Modèle de boîte



Positionnement statique : *static*

- Valeur par défaut,
- les éléments apparaissent dans le même ordre que dans le code *HTML*,
- cet ordre est appelé le **flux** de la page *HTML*,
- impossible d'utiliser les valeurs : top, left, right, bottom.
- la largeur par défaut d'un type *block* est la largeur du conteneur.

Positionnement relatif : *relative*

- Les valeurs : top, left, right, bottom indiquent la position par rapport à celle que l'élément aurait du avoir dans le flux.
- ne modifie pas l'affichage des autres éléments dans le flux,
- garde les mêmes dimensions et marges que pour *static*.

Positionnement absolu : *absolute*

- l'élément est retiré du flux, l'affichage des autres éléments est effectué comme s'il n'existait pas,
- Par défaut, l'élément est affiché là où il aurait du être affiché dans le flux (risque de superpositions),
- on utilise donc les valeurs : top, left, right, bottom,
- ces valeurs indiquent la position du premier ascendant qui n'est pas statique (au pire, le navigateur),
- la largeur par défaut est la largeur minimum pour afficher le contenu.

Positionnement fixe : *fixed*

identique à *absolute* sauf pour deux choses :

- le positionnement se fait toujours par rapport aux bords du navigateur.
- en cas de défilement de la page, l'élément ne bouge pas.

Objets flottants

Un objet flottant que l'on peut placer à droite ou à gauche de son conteneur. Le contenu des autres éléments évite alors de se superposer avec l'élément flottant.

HTML : Classes et identifiants

Afin de marquer des balises comme possédant des caractéristiques communes ou unique, il est possible de leur attribuer une ou plusieurs classes, ou un identifiant.

Classes

Les classes permettent d'identifier un ensemble de balises.

Dans le code HTML

```
1 <article class="exemple">
2   <header<h2 class="italique">Les classes</h2</header>
3   Ici, contrairement au monde reel, il n'y a pas de lutte des <em class="italique , rouge">classes</em>
4 </article>
```

Dans le code CSS

```
1 .exemple{
2   padding:1%;
3   border: 1px black solid;
4   border-radius:25px;
5 }
6 .italique{
7   font-style:italic;
8 }
9 .rouge{
10  color:red;
11 }
```

Classes

Les classes permettent d'identifier un ensemble de balises.

Dans le code HTML

```
1 <article class="exemple">
2   <header<h2 class="italique">Les classes</h2</header>
3   Ici, contrairement au monde reel, il n'y a pas de lutte des <em class="italique , rouge">classes</em>
4 </article>
```

Dans le code CSS

```
1 .exemple{
2   padding:1%;
3   border: 1px black solid;
4   border-radius:25px;
5 }
6 .italique{
7   font-style:italic;
8 }
9 .rouge{
10  color:red;
11 }
```

Classes et identifiants

Classes

Les classes permettent d'identifier un ensemble de balises.

Dans le code HTML

```
1 <article class="exemple">
2   <header<h2 class="italique">Les classes</h2</header>
3   Ici, contrairement au monde reel, il n'y a pas de lutte des <em class="italique , rouge">classes</em>
4 </article>
```

Dans le code CSS

```
1 .exemple{
2   padding:1%;
3   border: 1px black solid;
4   border-radius:25px;
5 }
6 .italique{
7   font-style:italic;
8 }
9 .rouge{
10  color:red;
11 }
```

Identifiant

Les identifiants permettent d'identifier une balise unique.

Dans le code HTML

```
1 <div id="jolifond">
2   Ce bloc va avoir un joli fond rien que pour lui.
3 </div>
```

Dans le code CSS

```
1 #jolifond{
2   background-image:url("joli.jpg");
3   background-position: top left;
4   background-repeat:no-repeat;
5 }
```

Identifiant

Les identifiants permettent d'identifier une balise unique.

Dans le code HTML

```
1 <div id="jolifond">
2   Ce bloc va avoir un joli fond rien que pour lui.
3 </div>
```

Dans le code CSS

```
1 #jolifond{
2   background-image:url("joli.jpg");
3   background-position: top left;
4   background-repeat:no-repeat;
5 }
```

Identifiant

Les identifiants permettent d'identifier une balise unique.

Dans le code HTML

```
1 <div id="jolifond">
2   Ce bloc va avoir un joli fond rien que pour lui.
3 </div>
```

Dans le code CSS

```
1 #jolifond{
2   background-image:url("joli.jpg");
3   background-position: top left;
4   background-repeat:no-repeat;
5 }
```


Les pseudos-classes

Permet de spécifier un état particulier d'une balise.

Quelques exemples

Interaction avec la souris

- **:hover** la souris passe sur la balise.
- **:active** clic maintenu sur la balise.

Les liens hypertextes :

- **:link** lien non visité.
- **:visited** lien visité.

Les `<input>` des formulaires

- **:valid** les valeurs sont valides.
- **:invalid** les valeurs sont invalides.

Les pseudos-éléments

Permet de spécifier la mise en page de parties spécifiques d'une balise.

- : **:before** permet d'insérer du contenu avant celui d'une balise.
- : **:after** permet d'insérer du contenu après celui d'une balise.
- : **:selection** zone sélectionnée/surlignée par l'utilisateur.
- : **:first-line** première ligne de texte du conteneur.
- : **:first-letter** première lettre de texte du conteneur.

Combinaisons de sélecteurs

Il est possible de combiner plusieurs sélecteurs.

Conjonction

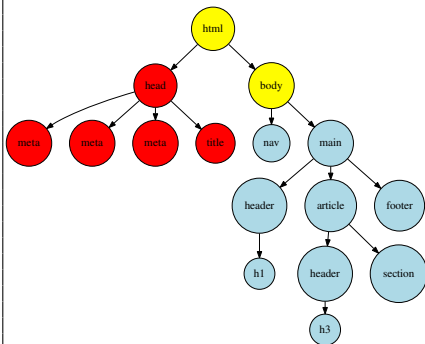
```
1 .lol::before, .lol::after{  
2     content:url("lol.png");  
3 }
```

Rappel : Le DOM

Code html

```
1 <!doctype htm>
2 <html>
3   <head>
4     <meta charset="UTF-8" />
5   </head>
6
7   <body>
8     <nav>
9       <a href="5.image.html">Image</a>
10      <a href="6.son.html">Son</a>
11      <a href="7.video.html">Video</a>
12    </nav>
13
14    <main>
15      <header>
16        <h1>Je suis l'en-tete principal de la page</h1>
17      </header>
18
19      <article>
20        <header>
21          <h3>Je suis l'en-tete d'un article</h3>
22        </header>
23        <section>
24          Je suis une section dans un article.
25        </section>
26      </article>
27
28      <footer>Je suis le pied de page de l' article.</footer>
29    </main>
30
31  </body>
32 </html>
33
34
```

DOM



Combinaisons de sélecteurs

$A B$	B descendant de A.
$A > B$	B descendant direct de A.
$A + B$	B directement adjacent à A (B après A).
$A \sim B$	B adjacent à A (B après A).
$A.B$	tous les A de classe B.
$A \# B$	A avec identifiant B.
A,B	A et B.

Héritage par défaut

Certaines propriétés sont héritées par défaut par tous les éléments du conteneur.

- La couleur du texte,
- La police,
- Les styles de listes,
- ...

Exceptions

Les balises ayant des propriétés prédéfinies n'hériteront pas automatiquement.

- la couleur des liens hypertextes,
- les modifications de la police pour les balises `<h1>`, ..., `<h6>`, `` et ``.

Héritage par défaut

Certaines propriétés sont héritées par défaut par tous les éléments du conteneur.

- La couleur du texte,
- La police,
- Les styles de listes,
- ...

Exceptions

Les balises ayant des propriétés prédéfinies n'hériteront pas automatiquement.

- la couleur des liens hypertextes,
- les modifications de la police pour les balises `<h1>`, ..., `<h6>`, `` et ``.

Héritage explicite

Pour toute propriété, on peut préciser explicitement qu'elles vont être héritées du conteneur.

Exemple

```
1 main{
2   margin:2%;
3   border:1px solid black;
4 }
5 p{
6   margin:inherit;
7   border:inherit;
8 }
```