
CSS, les bases

Le CSS, c'est quoi ?

CSS : feuille de style en cascade

Permettent de générer la présentation d'une page HTML : séparer la structure (HTML) de sa présentation (CSS)

Ensemble de règles stylistiques applicables à un, ou plusieurs documents HTML => gain de taille du fichier HTML

Facilite la mise à jour graphique, favorise l'accessibilité

Gestion des différents médias possible (print, screen, mobile etc.)

Couleur de texte, image de fond, style de police, menu à gauche ou à droite : c'est le CSS qui va contrôler tout ça !

La syntaxe de base

La syntaxe est composée de 4 éléments :

- Le **sélecteur** est l'élément sur lequel on applique les propriétés (balise HTML, id, classe, etc.)
- La **propriété** est l'effet que l'on va vouloir donner (ex couleur de texte, positionnement, couleur de fond, etc.)
- La **valeur** de la propriété CSS (rouge, 10px, etc.)
- Les **brackets** { }

Les commentaires CSS

Pour créer un commentaire CSS, on utilise `/* commentaire */`

Le commentaire n'est pas visible dans le navigateur

Sert au développeur à prendre des notes, laisser des messages aux autres et savoir à quoi servent les lignes de code

Le sélecteur usuel d'élément HTML

On peut sélectionner n'importe quel élément HTML et lui appliquer un style.

Une propriété appliquée sur un élément HTML, s'applique par défaut à tous les éléments présents dans le document HTML.

`p {color : blue; }` => Tous les paragraphes auront une couleur de texte bleu

Hiérarchie et généalogie

```
<div>
  <h1>Cupcake ipsum dolor sit</h1>
  <p>
    amet jelly sugar plum bear claw bonbon. Wypas dragée chocolate bar fruitcake. Gingerbread
    wypas muffin cotton candy toffee sugar plum I love danish icing.
  </p>
  <h2>Ice cream topping</h2>
  <p>
    <a href="#">Toffee gummi bears candy canes tiramisu bear</a>
    claw candy canes chocolate bar candy.
    <strong>Ice cream topping apple pie</strong>Cotton candy jelly-o gummies.
    <em>I love lollipop I love jelly</em>
  </p>
</div>
```

div

h1

p

h2

p

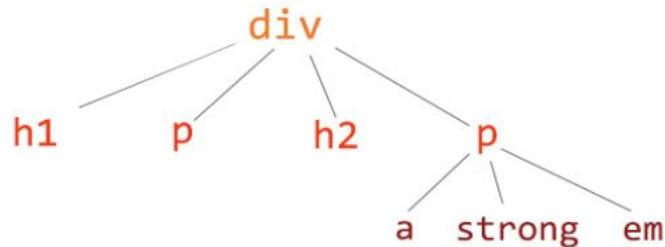
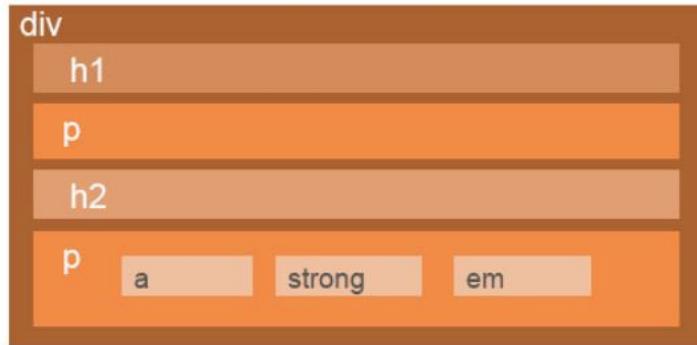
a

strong

em

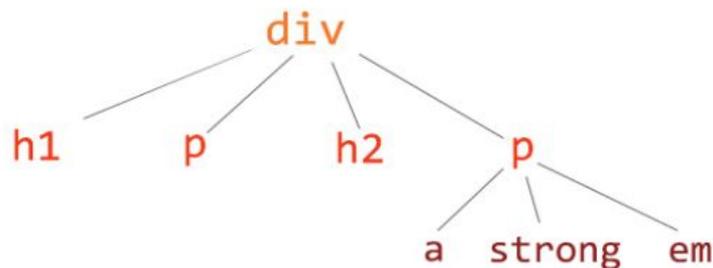
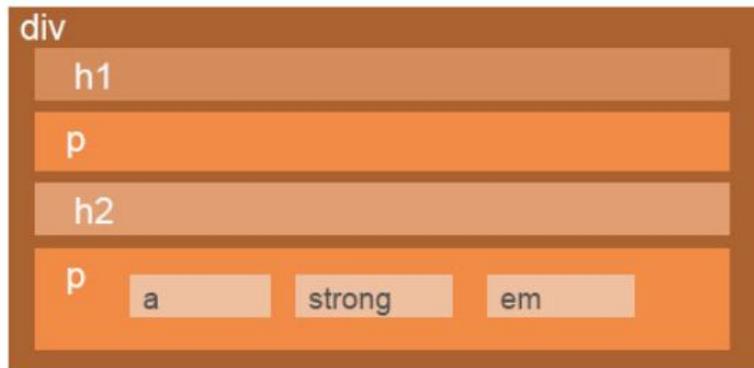
Notion d'enfant et de descendance

- h1, p, h2, p sont enfants de div
- a, strong et em sont enfant du p dans lequel ils sont contenus (mais pas de l'autre p)
- a, strong et em (et h1, p, h2 et p) sont descendants de div



Notion de parent et d'ancêtre

- div est **parent** de h1 p, h2, p
- Le 2ème p est **parent** de a, strong et em.
- div est **ancêtre** de a, strong et em (et de h1, p h2 et p)



Sélecteur de hiérarchie

Pour sélectionner le **a** descendant de **p**, nous allons pouvoir écrire :

p a { ... }

(attention à l'espace entre le p et le a)

Sélecteur de groupe

Pour sélectionner plusieurs éléments et leur appliquer la même valeur, on les sépare par une virgule.

`h1, h2 { color: red; }` => Va donner la couleur rouge à tous les h1 ou h2

Sélecteurs de classe

`class=" "` est un attribut qui peut se mettre sur n'importe quelle balise

Il permet de cibler de manière plus « spécifique » certains éléments

`<p class="important">` Un paragraphe spécifique qui est mis en avant `</p>`

Cibler une classe indépendamment de la balise

Si on veut cibler une classe, sans se préoccuper de la balise on utilise le « point ».

```
.nomdeclasse { }
```

Par exemple `.remarque { }` va cibler

```
<p class="remarque"> ... </p>
```

```
<h2 class="remarque"> ... </h2>
```

Une même classe peut être utilisée sur plusieurs balises HTML

Cibler une balise HTML avec une classe

Pour cibler un élément doté d'une classe en CSS on procède de la manière suivante (notez qu'il n'y a pas d'espace)

```
element.maclasse { }
```

Par exemple : `p.cefaid{ }` va cibler tous les éléments comme celui-ci :

```
<p class="cefaid"> ... </p>
```

L'id (identifiant)

Il a le même rôle qu'une classe, mais doit être unique sur la page (donc on va moins l'utiliser)

Un seul attribut id par balise est possible

On le note `<element id="monid">`

On le cible en CSS avec `#monid { }`

Espace ou pas ?

Pas d'espace signifie que toutes les conditions DOIVENT être vraies !

`p.cefaid{ }` cible tous les paragraphes **AYANT** une classe « cefaid »

```
<p class="cefaid"> ... </p>
```

Espace ou pas ?

Un espace entre deux « sélecteurs » détermine la généalogie :

`p .cefaid { }` cible tous les éléments qui ont pour classe « cefaid » et qui sont descendants d'un paragraphe

```
<p> ... <a class="cefaid"> ... </a> </p>
```

```
<p> ... <span class="cefaid"> ... </span> </p>
```

```
<p> ... <em class="cefaid"> ... </em> </p>
```

...

Attention!

Un nom de classe ou d'ID ne doit pas avoir d'accent ou caractères spéciaux

On a tendance à éviter le « tout majuscule »

Un nom d'ID ne prend jamais d'espace

Un nom de classe ne prend pas d'espace, sinon c'est une nouvelle classe

CSS	HTML ciblé
<code>p {}</code>	<code><p> </p></code>
<code>.maclasse {}</code>	<code><element class="maclasse"> ... </element></code>
<code>#monid {}</code>	<code><element id="monid"> ... </element></code>
<code>p.maclasse {}</code>	<code><p class="maclasse"> ... </p></code>
<code>p .maclasse {}</code>	<code><p> <element class="maclasse"> </element> </p></code>
<code>p, div {}</code>	<code><p> ... <p> <div> ... </div></code>

Pour vous entrainer:
<http://flukeout.github.io/>