

CSS3 – LES BASES

TABLE DES MATIÈRES

Syntaxe du CSS	5
Où placer son code CSS ?	7
Dans un fichier séparé	7
Dans la section <head>	8
Dans l'attribut "style"	9
Les sélecteurs simples	9
Sélecteurs d'éléments	9
Changer le comportement par défaut des éléments	9
Sélecteurs de classe	10
Sélecteurs sur base de l'id	11

Historique des versions

Version	Date	Modification
1.0.0	23/11/21	Création

NOTE : dans sa version actuelle, ce cours est très fortement inspiré/aspiré du cours « Apprendre à coder en HTML et CSS | Cours complet (2020) » de Pierre Giraud », accessible à l'adresse <https://www.pierre-giraud.com/html-css-apprendre-coder-cours>.

J'y ai apporté quelques modifications, ajouts et simplifications afin de mieux correspondre à notre contexte d'apprentissage.

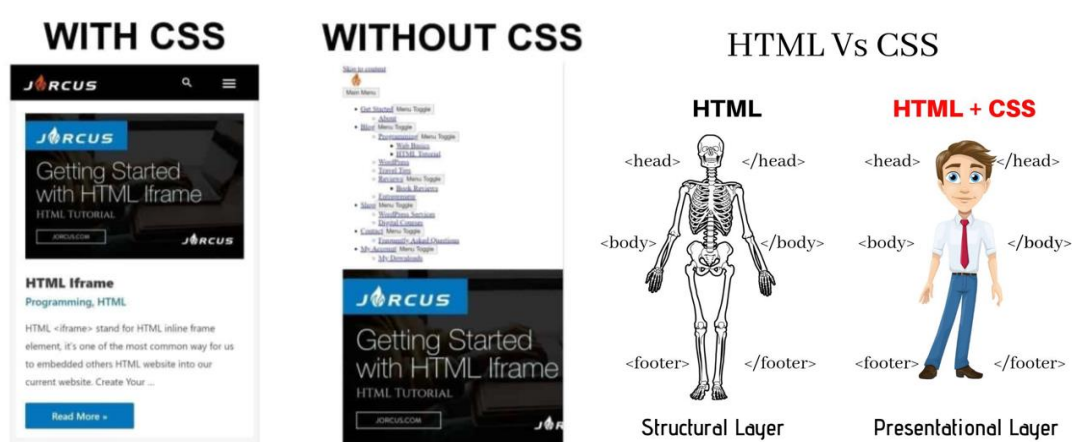
Au fur et à mesure des versions, le contenu sera rendu plus spécifique.

Comme on l'a vu dans l'introduction du cours, alors que le HTML permet de définir et structurer le contenu d'une page HTML, **le code CSS** permet de **modifier et gérer l'apparence visuelle**¹ du rendu généré par le navigateur.



CSS veut dire « **cascading style sheets** » ou, en français, « **feuilles de style en cascade** ».

Ce sont donc des **styles** que tu peux **appliquer** à des éléments HTML afin d'en **changer l'apparence** (couleur des caractères, couleur de fond, police, gras, taille...), **le positionnement** et/ou créer des **animations**.



Autrement dit, le code CSS va permettre d'habiller et styliser le contenu HTML.

SYNTAXE DU CSS

CSS est un langage **basé sur des règles** — on définit des règles de styles **destinées à des éléments ou des groupes d'éléments** particuliers dans la page.

Par exemple, imagine que je veuille que le titre principal de ma page s'affiche en rouge en gros caractères. Dans le code suivant, une règle CSS élémentaire réalise cette mise en forme :

```
h1 {
  color: red;
  font-size: 5em;
}
```

¹ Oui, c'est un pléonasme, qui permet d'insister sur le côté... visuel ;-)

La **règle** commence par un **sélecteur**, qui permet de sélectionner spécifiquement quels sont les **éléments**² auxquels le style va s'appliquer. Ici le style s'applique aux titres de niveau 1 (<h1>). Tu verras qu'il existe **plusieurs types de sélecteurs**, que nous allons découvrir au fur et à mesure.

Règle

```
h1 {  
  color: red;  
  font-size: 5em;  
}
```

Suivent une **paire d'accolades** `{ }` qui définissent un **bloc de déclaration** à l'intérieur duquel on trouve **une ou plusieurs déclarations**.

Bloc de déclaration

```
h1 {  
  color: red;  
  font-size: 5em;  
}
```

Une **déclaration** est constituée d'une paire **propriété : valeur**. Chaque paire précise une propriété de l'élément sélectionné, suivie de la valeur choisie pour cette propriété ; la propriété et la valeur sont séparées par deux points (:). Chaque déclaration **se termine par un point-virgule**. À chaque propriété définie par CSS correspondent différentes valeurs possibles.

Sélecteur

Déclaration

```
h1 {  
  color: red;  
  font-size: 5em;  
}
```

Propriété

Valeur(s)

² Souviens-toi de ce qu'est un **élément** en HTML : il est représenté par une/des balise.s et un contenu.

Dans l'exemple avec le `h1` ci-dessus:

- `h1` est le **sélecteur**
- `color` et `font-size` sont des **propriétés**
- `red` et `5em` sont les **valeurs** associées aux propriétés.

Une feuille de style CSS est constituée d'une **succession de règles**, par exemple :

```
h1 {  
  color: red;  
  font-size: 5em;  
}  
  
p {  
  color: black;  
}
```

On retient facilement certaines valeurs, d'autres sont plus difficiles à mémoriser. Certaines seront utilisées plus souvent que d'autres. C'est comme ça. Pour ces raisons, il est toujours utile d'avoir des sites de référence sous la main (w3school, mdn...).

OÙ PLACER SON CODE CSS ?

Le code CSS peut être mis dans 3 endroits différents.

- Dans l'**attribut `style`** d'un élément HTML
- Dans la **balise `<style>`** dans l'élément `<head>`
- Dans **un fichier `.css` séparé** ← **Méthode préférée !!!**

DANS UN FICHIER SÉPARÉ

Allons droit au but : **le meilleur endroit pour écrire ton code CSS, c'est dans un fichier `.css` séparé**, que tu pourras référencer dans toutes tes pages. C'est **facile**, c'est **propre** et c'est **efficace**, parce que ça évite de faire des copier-coller et les règles sont **centralisées** au même endroit. Du coup, c'est **plus facile à maintenir**.

Comment faire ? **Écris tes règles css dans un fichier séparé**, qui aura `.css` comme extension. Par convention, ce fichier est souvent placé dans un répertoire « style » ou « css ». Ça aura son utilité si tu crées plusieurs fichiers css pour ton site. Oui, c'est

possible, et c'est même parfois recommandé. On verra quand et pourquoi plus loin dans le cours.

Ensuite, il suffit de référencer ton fichier css dans la section `<head>` des pages html qui en ont besoin :

```
<head>
  <title>Titre de la page</title>
  <meta charset="utf-8">

  <link rel="stylesheet" href="css/styles.css">

</head>
```

Cet élément `<link>` indique au navigateur la présence d'une feuille de style, grâce à l'attribut `rel` ; la valeur de l'attribut `href` donne la localisation du fichier CSS.

Note : dans le fichier css, il ne faut pas ajouter la balise `<style>`, seulement les règles.

DANS LA SECTION <HEAD>

C'est la façon qu'on a utilisée dans l'introduction du cours, parce que c'est direct et facile. Par contre, cette méthode atteindra sa limite dès que tu ajouteras une 2^e page à ton site qui nécessite les mêmes règles de style.

Néanmoins, dans certains cas, on peut **à la fois** référencer une feuille de **style externe** comme on l'a fait dans le paragraphe précédent **ET** ajouter **une balise** `<style>` dans le fichier html pour ajouter ou écraser les styles de la feuille externe. C'est technique et on verra le pourquoi du comment plus tard.

```
<head>
  <style>
    h1 {
      color: red;
      font-size: 5em;
    }

    p {
      color: black;
    }
  </style>
</head>
```

DANS L'ATTRIBUT "STYLE"

Il est possible d'ajouter un attribut `style` à tout élément html, directement dans la balise ouvrante. À utiliser avec parcimonie, sinon tu arriveras rapidement à un code difficile à maintenir, avec plein de règles redondantes.

Mais, ponctuellement, cela peut rendre service pour modifier le style localement et changer/annuler une règle qui aurait été créée par ailleurs.

Exemple :

```
<h1 style="color: red; font-size: 5em">Développeur web</h1>
```

LES SÉLECTEURS SIMPLES

Pour rappel, les sélecteurs permettent de... **sélectionner les éléments** (d'où leur nom) auxquels on veut appliquer une série de déclarations css.

On va commencer par les cas simples. Plus tard, non verrons comment les combiner, et comment utiliser des sélecteurs avancés.

SÉLECTEURS D'ÉLÉMENTS

Comme leur nom l'indique, les **sélecteurs d'éléments** permettent de **cibler des éléments html spécifiques**.

Dans la règle CSS, le sélecteur correspond donc au nom d'un élément HTML. Par exemple, pour appliquer un style à tous les paragraphes du document HTML on utilisera le sélecteur `p`. Voilà la règle pour passer en vert tous les paragraphes :

```
p {  
  color: green;  
}
```

CHANGER LE COMPORTEMENT PAR DÉFAUT DES ÉLÉMENTS

Quand on observe un document HTML bien formé, même simple, on peut voir comment un navigateur le rend lisible par **une mise en forme par défaut**. Les titres sont écrits en gras dans une taille plus grande, les items des listes sont précédés d'une

puce. **Les navigateurs ont leurs feuilles de style internes** qu'ils appliquent **par défaut** à toutes les pages ; sans cela, tout le texte s'agglutinerait en paquet et il faudrait tout mettre en forme à partir de zéro. Tous les navigateurs modernes rendent les contenus HTML par défaut essentiellement de la même manière.

On recherche pourtant souvent autre chose que ce rendu par défaut. Il suffit alors de sélectionner l'élément HTML dont on veut modifier le rendu et d'écrire la règle CSS pour réaliser cette mise en forme.

Il est par exemple possible de changer l'apparence d'une liste non-ordonnée :

<pre> Timothée Chalamet Zendaya Oscar Isaac Rebecca Fergusson </pre>		<ul style="list-style-type: none">• Timothée Chalamet• Zendaya• Oscar Isaac• Rebecca Fergusson
---	--	---

Par défaut les navigateurs affichent des puces rondes et pleines. Grâce aux feuilles de style, il est possible de changer ce comportement / cette apparence :

<pre>li { list-style-type: circle; }</pre>		<ul style="list-style-type: none">○ Timothée Chalamet○ Zendaya○ Oscar Isaac○ Rebecca Fergusson
--	--	---

SÉLECTEURS DE CLASSE

Jusqu'ici, nous avons mis en forme des éléments HTML repérés par leur nom de balise. Cela fonctionne tant que vous voulez appliquer le même style à tous les éléments de ce type dans le document. La plupart du temps ce n'est pas le comportement désiré ; il faut donc trouver **une méthode pour sélectionner un sous-ensemble des éléments** à mettre en forme sans changer l'apparence des autres éléments du même type. L'approche la plus commune pour obtenir ce comportement est d'ajouter une **classe** (pensez à une étiquette) aux éléments HTML à mettre en forme puis de sélectionner cette classe en css.

Pour ajouter une classe à un élément, on ajoute l'attribut **class** dans la balise ouvrante. Exemple :

```
<li class="girl">Zendaya</li>
```

Il suffit ensuite d'ajouter une règle dans ta feuille de style qui s'appliquera uniquement aux éléments qui possèdent cette classe.

Le sélecteur de classe s'écrit en précédant le nom de la classe par un point (.) :

```
.girl {  
    color: pink;  
}
```

Grâce à cette règle, tous les éléments de la page qui possèdent la class « girl » seront affichés en rose³. Exemple :

```
<h1 class="girl">Casting</h1>  
<ul>  
    <li>Timothée Chalamet</li>  
    <li class="girl">Zendaya</li>  
    <li>Oscar Isaac</li>  
    <li class="girl">Rebecca Fergusson</li>  
</ul>
```

Casting

- Timothée Chalamet
- Zendaya
- Oscar Isaac
- Rebecca Fergusson

SÉLECTEURS SUR BASE DE L'ID

Comme vu dans le cours sur html, chaque élément peut avoir un attribut `id`, qui identifie l'élément sur la page. La valeur de cet élément doit être unique dans une même page.

Le sélecteur qui permet d'appliquer une règle css à un élément portant un `id` spécifique est **précédé par un hashtag (#)**.

```
#header {  
    background-color: yellow;  
}
```

³ Bonjour les clichés, désolé :-/