

---

# Développement

## CSS

Jason Le Scour

---

---

### Sommaire

Développement	1
CSS	1
<b>Sommaire</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
<b>Emplacement de styles</b>	<b>3</b>
<b>Feuille de style</b>	<b>4</b>
<b>Les sélecteurs</b>	<b>5</b>
<b>Les classes</b>	<b>6</b>
<b>Ordre de priorité de style</b>	<b>9</b>
<b>Positionnement de blocs</b>	<b>10</b>

---

# Introduction

## Signification de CSS

CSS, signifie "Cascading Style Sheets", ce qui se traduit par feuilles de styles en cascade. La feuille de style fournit la mise en forme des éléments de la page, qui auront été écrits en HTML.

Le terme "en cascade" indique que la mise en forme d'une page peut faire appel à plusieurs feuilles de style.

Les différentes propriétés affectées à un même élément s'ajoutent alors pour lui donner sa mise en forme finale.

Lorsque deux propriétés se contredisent, des règles de priorité s'appliquent et c'est généralement le dernier style défini qui est pris en compte.

Voici comment sélectionner un élément de la page et lui attribuer une propriété de mise en forme.

### Principe :

une règle de style comprend :

- Un **sélecteur** : il s'agit des balises concernées par cette règle.
- Un **bloc de déclaration** : il indique les propriétés à attribuer à ces balises.
- Un **bloc de déclarations** : il indique les propriétés à attribuer à ces balises.

Chaque déclaration est du type : **propriété : valeur;**

Exemple :    `h3 ( font-style: italic; font-family: Arial, sans-serif; )`

	1ère	2ème
Sélecteur	déclaration	déclaration

Cette règle indique que les titres de niveau 3 (code html encadrés par `<h3> ... </h3>`) s'afficheront en italique et en Arial (ou dans une police générique sans-serif si la police Arial est absente).

**Remarque :** une règle peut s'écrire sur plusieurs lignes :

```
h3 {  
    font-style: italic;  
    font-family: Arial, Sans-Serif;  
}
```

---

## Emplacement de styles

### Les règles de style peuvent se trouver :

- Dans le code HTML, comme attributs de balises : ce sont des style en ligne (utilisation déconseillée);
- Dans l'en-tête de la page web : feuille de style interne;
- Dans un fichier distinct : feuille de style externe, à appeler dans l'en-tête de la page web (le top).

---

# Feuille de style

## Feuille de style interne

Lorsque les règles de style sont regroupées dans l'en-tête de la page web, elles constituent une feuille de style interne.

Les styles sont écrits entre balises `<head>` et `</head>`, à l'intérieur d'une balise `<style>` :

```
<head>
...
<style type="text/css">
<!-- ... règles de styles ici ... -->
</style>
...
</head>
```

## Feuille de style externe

Lorsque des règles de styles sont applicables à plusieurs pages web, il est intéressant de les écrire dans un fichier à part. Cette feuille de style externe est appelée par chacune des pages concernées.

### Avantages :

- L'unité graphique du site.
- Facilite les modifications (passe du bleu à l'orange par exemple).

Une feuille de style externe est un fichier d'extension `.css` :

- C'est un fichier texte qui contient l'ensemble des règles définies.
- Il ne contient pas de balises `<style> ... </style>`
- Les commentaires sont entre `/*` et `*/`

Pour que cette feuille de style soit prise en compte dans une page web, il suffit de l'appeler dans l'en-tête :

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

---

## Les sélecteurs

Ils permettent d'appliquer un style à un élément. On peut définir un style propre à chaque élément comme il existe par défaut dans les navigateur.

Exemple :

- `p { color: white; background-color: blue; }`

Pour que le texte de tous les paragraphes figure en blanc sur fond bleu

### Le sélecteur universel

Pour appliquer un style à tous les éléments, on utilise le sélecteur universel `*` avant la définition d'une ou plusieurs propriétés.

Exemple :

- `* { background-color: yellow; }`
- `p { background-color: gray; }`

La couleur de fond de tous les éléments est en jaune, sauf celle des paragraphes qui est en gris.

### Les salvateurs multiples

Ils permettent d'appliquer un style à plusieurs éléments.

Exemple :

- `h1 { color: black; background-color: red; }`
- `h2 { color: black; background-color: red; }`
- `p { color: black; background-color: red; }`

Peut s'écrire :

- `h1, h2, p { color: black; background-color: red; }`

**Remarque :** pour des paragraphes avec une Marge on ajoute :

- `p { margin: 20px; }`

---

## Les classes

Il est possible d'appliquer les mêmes propriétés à plusieurs éléments de la page web, quelle que soit la balise qui les entoure.

Il suffit de leur attribuer une classe commune en utilisant l'attribut `class` des éléments HTML.

Pour créer une classe en CSS, le sélecteur est constitué du nom choisi pour la classe précédé d'un point (.).

Le nom d'une classe peut être un mot quelconque, en évitant quand même les noms des propriétés CSS et des éléments XHTML car cela occasionnerait des confusions.

### Exemple :

- **HTML**

```
<p class="menu">
<h1 class="menu">
<h1 class="menu">
```

- **CSS**

```
.menu {
    font-family : Arial;
    font-style : italic;
}
```

### Exemple 2 :

- **HTML**

```
<table class="bordure">
<tbody>
<tr>
<td class="bordure"> </td>
```

- **CSS**

---

En HTML, une alternative aux classes est utilisée pour repérer un élément unique dans une page web. Il s'agit de donner à la balise un **identifiant** :

```
<div id="maDivision"> ... </div>
<p id="mon Paragraphe"> ... </p>
```

Ce qui permet de préciser cet identifiant dans la feuille de style, à l'aide du caractère dièse # :

```
Div#maDivision {
Text-align right;
}
```

```
p#monParagraphe {
text-align left;
}
```

	Classe	Identifiant
Balise	<p class="toto">	<p id="toto">
Règle de style	p.toto ... .toto ...	p#toto ... #toto ...
Éléments concernés	Plusieurs balises, identiques ou différentes.	Une seule balise dans toute la page

## Les pseudo-classes

Les pseudo-classes accolées à une balise, apportent des précisions aux sélecteurs. Cette méthode permet d'écrire des propriétés à utiliser uniquement dans certains cas de figure.

Les principales pseudo-classes sont les suivantes :

Les pseudo-classes	Leurs rôles
:link	• lien hypertexte qui n'a pas été visité;
:visited	• lien visité;
:hover	• élément survolé par la souris;
:active	• élément activé

**Remarque** : La pseudo-classe la plus utilisée est `hover`; elle indique que la règle de style n'est à appliquer qu'au passage de la souris.

---

## Les pseudo-éléments :

Les pseudo-éléments apportent d'autres types de précision :

Les pseudo-éléments	Leurs rôles
:first-letter	• Première lettre du bloc
:first-line	• Première ligne du bloc
:before	• Avant la balise spécifié
:after	• Après la balise spécifié

L'exemple suivant agrandit la taille de la première lettre pour chaque paragraphe <p>

---

## Ordre de priorité de style

Si une règle de style vient contredire une règle précédente, c'est en général le dernier style défini qui s'applique.

### Règle de style prioritaire

Pour qu'un style ne soit pas modifié par un autre, il faut écrire **!important** avant le point virgule qui termine la propriété.

#### Exemple :

- `body { background-color: white !important; }`

---

## Positionnement de blocs

Les feuilles de style nous proposent une méthode claire pour la mise en page : elle consiste à positionner les éléments dans la page ou dans leur bloc conteneur. Toutefois, il faudra au préalable définir les dimensions et les marges, intérieures et extérieures, de chaque élément.