



LA MANU

ÉCOLE SUPÉRIEURE  
DES MÉTIERS DU NUMÉRIQUE

# LE LANGAGE CSS

Découvrir le langage CSS : son rôle, sa syntaxe.

# INTRODUCTION AU LANGAGE CSS

# Le langage CSS

CSS (Cascading Style Sheets) signifie feuilles de style en **cascades**.

La cascade correspond au rendu final des différentes sources de styles appliquées à un même document.

Le CSS décrit la façon dont les navigateurs doivent effectuer le **rendu visuel** des éléments HTML sur écran, papier ou d'autres médias.

Le CSS permet de gérer la mise en page de nombreuses pages web simultanément.

Les feuilles de style externe sont enregistrées dans des fichiers «**.css** ».



# Principes de fonctionnement du CSS : les sélecteurs

CSS permet d'attribuer des styles à des éléments HTML qu'il faut pouvoir **cibler**.

Le langage CSS passe par le choix pertinent de différents **sélecteurs** : éléments HTML, attributs, états, combinaisons de sélecteurs...

```
/* Ex : sélecteur d'élément */  
h1 {  
    font-size: 30px;  
    font-weight: bold;  
}  
  
/* Ex : sélecteur complexe */  
.product-box button:hover {  
    background-color: blue;  
    color: white;  
}
```

# Principes de fonctionnement du CSS : les déclarations CSS

Pour modifier les styles des éléments HTML, il faut déclarer des règles CSS.

Une règle CSS est constituée d'un **sélecteur** et d'un bloc de déclaration délimité par des « **{ ... }** » dans lequel chaque déclaration est séparée de la suivante par « **;** »..

Chaque déclaration fait intervenir une **propriété CSS** et sa **valeur** associée. La propriété et sa valeur sont séparées par un « **:** ».

Plusieurs sélecteurs peuvent partager le même bloc, si on les sépare par des virgules « **,** ».

```
selecteur-unique {  
    declaration : valeur;  
    declaration : valeur;  
}
```

```
selecteur-1,  
selecteur-2 {  
    declaration : valeur;  
    declaration : valeur;  
}
```

# Où écrire les règles CSS : le style « en ligne » avec l'attribut style

Dans un élément HTML à l'aide d'un attribut style :

```
<span style="color: blue;">
```

Mon texte en bleu

```
</span>
```

Pratique à exclure pour plusieurs raisons :

- Le style qui est défini de cette façon ne peut pas s'appliquer à d'autres éléments, dans sa page ou dans une autre. On perd en souplesse de maintenance (imaginez que vous deviez reprendre ce style sur 50 pages...).
- Les **styles « en ligne » sont prioritaires** sur les styles déclarés par les autres méthodes. Pour pouvoir surcharger un style « en ligne », il faudra ajouter « **!important** » à la fin de la déclaration.

Il faut toujours **éviter au maximum** d'en arriver là.

# Où écrire les règles CSS : la feuille de style interne avec `<style>`

L'élément `<style>` s'insère en priorité dans l'en-tête (**balise head**) d'une page web.

✗ Peu recommandé.

L'usage de la balise `style` est à éviter si possible. Il ne sera pas réutilisable d'une page à l'autre, ce qui rend la maintenance des pages web plus laborieuse.

```
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    body {
      background-color: white;
      color: black;
    }
  </style>
</head>
```

# Où écrire les règles CSS : la feuille de style externe

C'est LA bonne pratique !

- ✓ Réutilisable
- ✓ Facilité de maintenance

On lie le fichier css à la page web à l'aide de `<link>` dans le `<head>`.

Attention : ne pas oublier de préciser le type de relation : « **rel=stylesheet** ».

La valeur de « **href** » est une **URL absolue** ou une **URL relative** à la page web.

```
<head>  
  <meta charset="UTF-8">  
  <title>Document</title>  
  <link rel="stylesheet" href="css/  
s/styles.css">  
</head>
```



# Focus sur les sélecteurs

# Le sélecteur universel « \* »

Ce sélecteur s'applique à **tous les éléments HTML** d'une page ou d'un parent.

**Remarque** : le sélecteur universel est souvent utilisé dans une démarche de « **reset css** » qui permet d'annuler les styles des navigateurs.

```
* {  
    margin: 0;  
    padding: 0;  
}
```

# Les sélecteurs d'éléments ou sélecteurs simples

Les sélecteurs d'éléments ont une syntaxe qui est provient directement de la **balise** qui leur est attribué en langage HTML.

Le sélecteur « **table** » va cibler tous les éléments HTML <table>, sans aucune exception.

✓ Ils consomment **moins de ressources** pour être interprétés que leurs homologues complexes.

✓ Leur syntaxe est **plus claire** que les sélecteurs complexes.

✗ Ils manquent de **précision** pour des usages plus ciblés.

```
p {  
    color: #313131;  
}  
  
table {  
    width: 100%;  
    border-  
color: #cccccc;  
    border-  
style: solid;  
    border-  
width: 1px;  
}
```

# Le sélecteur d 'éléments descendants « espace »

Ce sélecteur permet de cibler tous les éléments <B> qui sont enfants d'un élément parent <A>.

## Remarques :

Plus il y a d'éléments enfants, plus le sélecteur gagne en spécificité.

Les éléments sélectionnés ne sont pas nécessairement des enfants directs du dernier parent indiqué dans le sélecteur.

```
/* Cible l'ensemble  
des <h1> contenus dans une <section> */  
  
section h1 {  
    color: blue;  
}  
  
/* Cible l'ensemble des <td> contenus  
dans des <tfoot> de tous les <div> */  
  
div tfoot td {  
    background-color: #e0e0e0;  
}
```

# Sélecteur d'attribut « id »

La valeur d'un attribut « **id** » doit être **unique** sur la page où il est utilisé. Par abus de langage, on dira qu'un seul élément peut recevoir le même « id ».

Tous les sélecteurs qui font référence à la valeur d'un attribut « id » s'écrivent avec un dièse « # » suivi de la valeur de « id ».

La spécificité des sélecteurs d'attribut « id » est supérieure à celle des sélecteurs d'attribut « class ».

```
.text-blue {  
    color: blue;  
}  
#must-be-red {  
    color: red;  
}
```

```
<p id="must-be-red" class="text-blue">Je suis rouge</p>
```

# Sélecteur d'attribut « class »

L'attribut « **class** » fait partie des **attributs universels** des éléments HTML. Ils permettent d'attribuer des **valeurs arbitraires** aux éléments de façon à mieux cibler les éléments en CSS.

La même valeur d'un attribut « **class** » peut être attribuée à plusieurs éléments HTML. Par abus de langage, on dira que plusieurs éléments peuvent avoir la même « class ». Tous les sélecteurs qui font référence par la valeur d'un attribut « class » s'écrivent avec un point « . », suivi de la valeur de « class ».

```
<p class="text-blue">Je suis bleu</p>
```

```
.text-blue {  
    color: blue;  
}
```

# Syntaxe chaînée

La syntaxe chaînée permet d'apporter de la **spécificité** dans le cas où un même élément peut être ciblé avec plusieurs sélecteurs.

Pour créer un sélecteur en syntaxe chaînée, on colle les sélecteurs les uns aux autres.

```
<h1 class="main-title underlined">Titre</h1>
```

```
<style>  
  h1.main-title.underlined {  
    /* styles */  
  }  
</style>
```

```
<span id="cart-total" class="big">30,00 €</span>
```

```
<style>  
  #cart-total.big {  
    /* styles */  
  }  
</style>
```

# Quelques autres sélecteurs d'attributs

Pour sélectionner un élément ayant un **attribut** « **attribut** », il faudra écrire :

```
[attribut] {  
    /* Déclarations de styles */  
}
```

Pour sélectionner un attribut ayant LA valeur « **foo** » alors on écrira

```
[attribut="foo"] {  
    /* Déclarations de styles */  
}
```

```
/* Cible tous éléments  
ayant un attribut name */  
[name] {  
    border-radius: 3px;  
}  
  
/* Cible tous éléments ayant  
un attribut name="user-pwd" */  
[name="user-pwd"] {  
    background: lightgrey;  
}
```