# CSS

Les CSS (Cascading Style Sheets, feuilles de style en cascade) ont pour rôle de présenter le contenu HTML. Elles permettent de séparer le contenu (HTML), de la présentation (CSS).

## Intégration

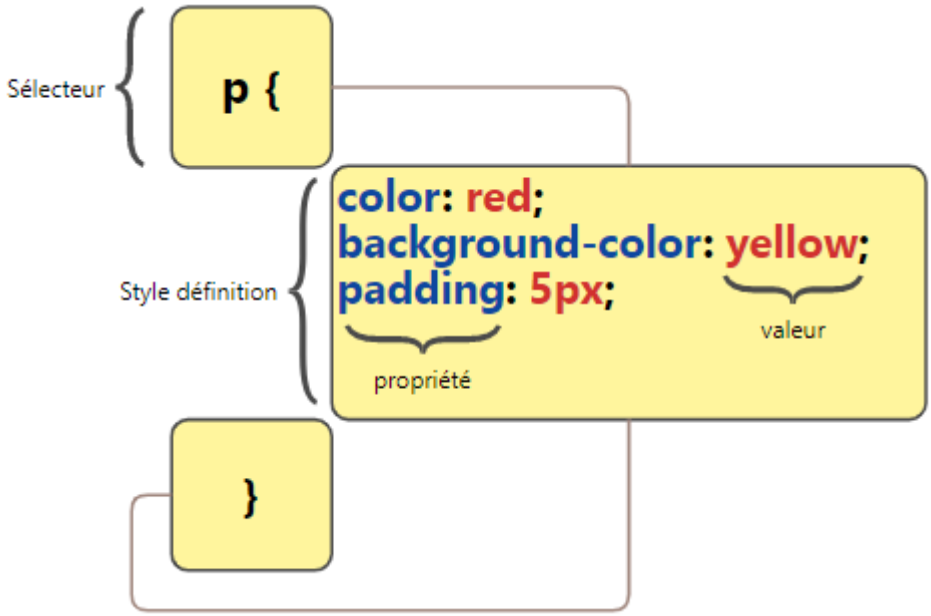
Intégration d'une feuille de style dans une page web, avec la balise **link**, dans la partie **head** :

```
<html>
<head>
    <meta charset="utf-8">
    <title>Page HTML avec Css associé</title>
    <link rel="stylesheet" href="css/styles.css">
</head>
...
</html>
```

## Principe

Définir un style consiste a associer une présentation :

- grâce à la définition d'un ensemble de propriétés,
- appliquées à des cibles HTML identifées par un sélecteur.




Association d'un texte de couleur rouge aux balises **h1** :

```
h1 {
```

```
  color: red;
}
```

### Héritage

Chaque élément enfant reçoit le style de son parent :
La font **sans-serif** va également s'appliquer à l'élément **h1** qui appartient au **body**

```
body {
  font-family: Verdana, Arial, Helvetica, sans-serif;
}

h1 {
  color: red;
}
```

Pour approfondir, voir Cascade et héritage

### Imbrication

Il est possible de définir explicitement le style d'éléments imbriqués :

Ce style s'applique a tous les **input** appartenant à une **div** :

```
div input {
  color: green;
}
```

## Sélecteurs

Le sélecteur CSS permet de définir le ou les éléments auquels un ensemble de styles va s'appliquer.

### Types de sélecteurs

#### Eléments

Application d'un style a un type d'élément (titres de niveau 1 en bleu):

```
h1 {
    color: blue;
}
```

#### Ids

Application d'un style a un élément défini par son id :

```
#elmId {
    color: blue;
}
```

#### Classe

Application d'un style a des éléments définis par une classe CSS :

```
.elmsClass {
    color: blue;
}
```

#### Attribut

Application d'un style a des éléments définis par leur type et la valeur d'un attribut :

```
a[href=index.html] {
    color: blue;
}
```

#### Pseudo élément

Permet de mettre en forme une partie d'un élément :

```
p::first-line {
  color: blue;
  text-transform: uppercase;
}
```

voir Pseudo éléments sur MDN

#### Pseudo classe

Permet de mettre en forme un élément en fonction de son état, par exemple le passage de la souris au dessus d'une div :

```
div:hover {
  background-color: #F89B4D;
}
```

voir Pseudo classes sur MDN

#### Liste de sélecteurs

Il est possible de cibler plusieurs sélecteurs, pour leur attribuer un même style :

```
.elmsClass, a, .elm {
    color: blue;
}
```

#### Chaînage

Elements **a** ayant pour classes **menu** et **item** avec une propriété **data-id** :

```
a.menu.item[data-id] {
    color: blue;
}
```

### Résumé

| Sélecteur CSS | Signification |
|---|---|
| * | Sélectionne tous les éléments |
| **E, F** | Sélectionne tous les éléments de type E et de type F |
| **E F** | Sélectionne tous les éléments F à l'intérieur des éléments E, quelque soit le niveau de descendance |
| **E > F** | Sélectionne les éléments F enfants directs des éléments E |
| **E + F** | Sélectionne tout élément F placé directement après un élément E |
| **E~F** | Sélectionne tout élément F placé après un élément E dans la page |

## Propriétés

Approfondir sur MDN

## Savoir-faire spécifiques

- Positionnement en CSS
- Media-queries

### Games

- Maîtriser les sélecteurs CSS avec Css diner
- Maîtriser le positionnement CSS avec Flexbox froggy

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
**http://slamwiki2.kobject.net/web/css**

Last update: **2023/03/26 19:57**