



# ***Web Academy***



## **Mise en forme de page web en CSS**

*Alexis EL MRINI (@alexis-elmrini)*  
*Tristan LE GODAIS (@PolariTOON)*

# Sommaire

I. Présentation des CSS .....	5
1. Vocation des CSS .....	5
2. Intégration dans un document .....	6
3. Bonnes pratiques .....	7
II. Syntaxe .....	8
III. Sélecteurs .....	9
1. Sélecteurs basiques .....	10

---

2. Sélecteurs d'attributs.....	11
3. Sélecteurs de fratrie (horizontaux) .....	12
4. Sélecteurs de filiation (verticaux).....	13
IV. Propriétés .....	14
1. Formatage de texte .....	14
2. Modèles de disposition .....	15
3. Modèle de boîte .....	19
4. Bordure et fond .....	20
V. Valeurs.....	21

---

1. Couleurs.....	22
2. Mesures .....	23

# I. Présentation des CSS

## 1. Vocation des CSS

Les *CSS (Cascading Style Sheets)* permettent de décrire le rendu d'un document en suivant le principe de séparation du fond de la forme :

### *HTML*

- ▶ Définit le fond (le message, le contenu)

### *CSS*

- ▶ Définissent la forme (la présentation, la mise en page)

## 2. Intégration dans un document

Il existe trois moyens pour intégrer une feuille de style CSS à un document *HTML* :

### Style *inline*

► `<div style="a: ...; b: ...;">...</div>`

### Feuille de style interne

► `<style>c {...} d {...}</style>`

### Feuille de style externe

► `<link rel="stylesheet" href="./style.css"/>`

### 3. Bonnes pratiques

Pour être bien sûr de séparer le fond de la forme, une méthode peut être de commencer par rédiger le contenu et ensuite de le mettre en page.

Dans cette optique, il est préférable d'utiliser des feuilles de styles externes !

## II. Syntaxe

Une feuille de style comporte une liste de règles, associant chacune un bloc de déclarations à une liste de sélecteurs :

```
sélecteur1 { /* Première règle */  
    propriété1: valeur1;  
    propriété2: valeur2;  
}  
sélecteur2, sélecteur3 { /* Seconde règle */  
    propriété3: valeur3;  
}
```

*Exemple de document CSS syntaxiquement valide*



## III. Sélecteurs

Les sélecteurs permettent de cibler les éléments concernés par la déclaration. Il en existe plusieurs types et il est possible de les combiner.

```
p#note, ul > li.short {...}
```

```
/*
```

```
  Cible l'élément <p id="note">...</p>
```

```
  et les éléments <li class="short ...">...</li>
```

```
  enfants d'éléments <ul>...</ul>
```

```
*/
```

*Règle utilisant une combinaison de plusieurs types de sélecteurs*

# 1. Sélecteurs basiques

## Sélecteur universel

- ▶ `* {...}` cible tous les éléments

## Sélecteur d'élément (spécificité 0.0.1)

- ▶ `tagname {...}` cible les éléments de nom `tagname`

## Sélecteur de classe (spécificité 0.1.0)

- ▶ `.classname {...}` cible les éléments de classe `classname`

## Sélecteur d'identifiant (spécificité 1.0.0)

- ▶ `#id {...}` cible l'élément d'identifiant `id`

## 2. Sélecteurs d'attribut

### Sélecteur de clé

- ▶ `a[attr] {...}` cible les éléments `a` possédant l'attribut `attr`

### Sélecteur d'association clé-valeur

- ▶ `a[attr=val] {...}` cible les éléments `a` possédant l'attribut `attr` ayant la valeur `val`

### 3. Sélecteurs de fratrie (horizontaux)

#### Sélecteur de suivant direct (cadet)

- ▶ `a + b {...}` cible le cadet `b` d'un élément `a`

#### Sélecteur de suivant généralisé

- ▶ `a ~ b {...}` cible les éléments `b` après un élément `a`

## 4. Sélecteurs de filiation (verticaux)

### Sélecteur de descendant direct (enfants)

- ▶ `a > b {...}` cible les enfants `b` d'un élément `a`

### Sélecteur de descendant généralisé

- ▶ `a b {...}` cible les éléments `b` dans un élément `a`

## IV. Propriétés

### 1. Formatage de texte

```
div {  
    color: red; /* couleur du texte */  
    font-family: Arial; /* police d'écriture */  
    font-style: italic; /* roman ou italique */  
    font-weight: bold; /* graisse */  
    text-decoration: underline; /* soulignage... */  
}
```

*Propriétés affectant le mise en forme d'un texte*

## 2. Modèles de disposition

CSS offre plusieurs modèles pour agencer le contenu d'un élément.

Plusieurs propriétés permettent d'activer les différents modèles : `display`, mais aussi `float` et `position`.

Remarquons que :

- `display` possède deux valeurs spéciales : `none` qui n'affiche pas l'élément et `contents` qui l'affiche sans générer de boîte
- Chaque modèle propose un ensemble de propriétés spécifiques pour décrire l'interaction entre les différents boîtes

## Flux

- ▶ C'est le modèle par défaut qui décrit comment interagit un élément avec ceux autour de lui

Au sein d'un tel élément, les éléments s'affichent dans l'ordre, dans le sens de la lecture (généralement de bas en haut pour les éléments *block* et de gauche à droite pour les éléments *inline*)

Dans ce modèle, cela correspond respectivement aux valeurs `block` et `inline` de la propriété `display`

- ▶ Il est possible de répartir le contenu d'un élément *block* en colonnes à l'aide des propriétés `column-*`
- ▶ Un élément peut sortir du flux si une des propriétés `float` et `position` est modifiée



## Boîte flottante

- ▶ Avec les propriétés `float` et `clear`, un élément peut flotter à gauche ou à droite en décalant le reste du flux

## Boîte positionnée

- ▶ Avec les propriétés `position`, `top`, `left`, `bottom` et `right`, la position d'un élément peut être définie très précisément en fournissant ses coordonnées

## Tableau

- ▶ C'est le modèle historiquement utilisé par les éléments `<table>...</table>`, `<tr>...</tr>`, `<th>...</th>`, `<td>...</td>`... avant qu'il ne soit défini au sein des CSS avec les valeurs `table` et `inline-table` de la propriété `display` et les propriétés `table-*`

## Boîte flexible

- C'est un modèle activable avec les valeurs `flex` et `inline-flex` de la propriété `display` conçu pour un *design* unidimensionnel

Le position, la taille et l'ordre des enfants se contrôle avec les propriétés `flex-*`, `place-*`, `*-gap` et `order`

## Grille

- C'est un modèle activable avec les valeurs `grid` et `inline-grid` de la propriété `display` conçu pour un *design* bidimensionnel

Le position, la taille et l'ordre des enfants se contrôle avec les propriétés `grid-*`, `place-*`, `*-gap` et `order`

### 3. Modèle de boîte

Un élément est lui-même constitué d'une imbrication de boîtes délimitées par la marge intérieure, la bordure, la marge extérieure et le décalage.

La dimension de chacune des boîtes peut notamment être contrôlée avec les propriétés `padding-*`, `border-**-width`, `margin-*`, `top`, `left`, `bottom` et `right`.

## 4. Bordure et fond

```
div {  
    border: 2px solid blue;  
    /* bordure */  
    border-right: 4px dotted red;  
    /* bordure droite */  
    background-color: black;  
    /* fond coloré */  
    background-image: url("../sprite.png");  
    /* fond imagé */  
}
```

*Propriétés affectant la bordure et le fond*

## V. Valeurs

Un certain nombre de propriétés accepte des types de valeurs qui sont récurrents, comme les longueurs et les couleurs.

À chaque type de valeurs correspond une série de notations aux effets divers.

# 1. Couleurs

Les couleurs peuvent être renseignées en utilisant :

## Des mots-clés

► `rebeccapurple`, `currentColor`, `transparent`...

## Une notation fonctionnelle

► `rgb(102, 51, 153)`, `rgba(102, 51, 153, .4)`...

## Une notation hexadécimale

► `#639`, `#663399`, `#6396`, `#66339966`...

## 2. Mesures

Chaque mesure est typée selon son unité. On trouve entre autres :

### Les entiers et les nombres

► 0, 1.25, -.4...

### Les pourcentages

► 25%...

### Les longueurs

► Absolues : px, mm, pt... ou relatives : em, vw, vh...