

C'est LA solution pour décorer une page Web de façon efficace.

Il existe des bibliothèques tierces qui permettent d'appliquer des décorations CSS prédéfinies.

Par exemple :

BOOTSTRAP : <https://getbootstrap.com/docs/5.2/getting-started/introduction/>

MATERIALIZE : <https://materializecss.com/>

Pour apprendre le CSS, le plus simple est d'aller sur un des nombreux sites de formation :

W3SCHOOLS : <https://www.w3schools.com/>

MOZILLA : <https://developer.mozilla.org/en-US/docs/Web/CSS>

ALSACREATION : <https://www.alsacreations.com/tuto/liste/2-css.html>

La description d'un CSS peut se faire :

- Dans un **document** .css indépendant : c'est la solution la meilleure
- Dans la zone <head> d'un document html, avec une **balise** <style> ,
- Ponctuellement dans un **attribut** style= dans une balise : <p style=" color:blue ;">

Installation recommandée :

Un document indépendant .css a été créé : on indique dans le document html le nom de la feuille de style utilisée.

Cela se fait dans la balise <head> :

```
<head>
    <link rel="stylesheet" href="ma_feuille.css" />
</head>
```

1. COMPOSITION DU DOCUMENT CSS

Syntaxe d'un réglage CSS :

Nom_du_paramètre : Valeur_du_réglage ;

Exemple : Pour provoquer le centrage du texte sur une ligne : text-align : center ;

Les sites de référence cités ci-dessus donnent une liste impressionnante des paramètres et réglages possibles.

SELECTEUR CSS : Affectation des réglages CSS

A quelle balise va s'appliquer le CSS qu'on aura écrit ? C'est là qu'intervient le **sélecteur CSS**.

La syntaxe générale est :

```
sélecteur {
    Nom_du_paramètre : Valeur_du_réglage ;
    Nom_du_paramètre : Valeur_du_réglage ;
    ...
}
```

Exemple : Pour appliquer un réglage commun à TOUTES les balises <div> d'un document :

```
div {
    text-align : center ;
    color : blue ;
    background-color : lightgrey ;
}
```

Différents sélecteurs simples

- Le sélecteur peut donc être une balise : div, input, p, a, ... ou même * pour désigner TOUTES les balises.

- Le sélecteur peut également être un attribut de balise :

#toto {...} toutes les balise avec un attribut id='toto'. Exemple : <p id=' toto' >Bonjour</p>

[attr] ou [attr=valeur] ; Toute balise avec un attribut spécifique et/ou une valeur spécifique

Exemple : [name=' login'] : une balise de type <input name=' login' />

Il existe de nombreuses autres combinaisons possibles avec les attributs. Voir les sites de référence.

- Le sélecteur peut être une CLASSE CSS : c'est une solution très utilisée. C'est le symbole « . » qui indique qu'il s'agit d'une classe CSS. Le « . » n'apparaît pas dans le code HTML. C'est l'attribut « class » qui est utilisé :

Exemple : Le CSS :

```
.titre {
    text-align : center ;
    color : blue ;
    background-color : lightgrey ;
}
```

Le HTML :

```
<div class="titre" >Mon site Web</div>
```

Plusieurs réglages pour un même élément HTML

Il est fréquent de fractionner un ensemble de réglages CSS en plusieurs classes CSS, ce qui permet de les réutiliser partiellement à plusieurs endroits.

Exemple : Une balise <div> avec un cumul de 3 classes CSS :

```
<div class="titre haut deco" >Mon site Web</div>
```

Il n'y a qu'un seul attribut « class », mais plusieurs classes CSS indiquées dans l'ordre d'exécution.

Sélecteurs CSS multiples

Un sélecteur pour plusieurs éléments :

```
div, .titre, p, a { ... }
```

On cible ici toutes les <div>, les balises avec un attribut « class='titre' », toutes les balises <p> et <a>

Le séparateur est la « , »

Un sélecteur pour trouver un élément enfant :

Exemple : Code HTML :

```
<div class="titre">
    <a href="page1.html"></a>
    <a href="page2.html"></a>
</div>
<a href="accueil.html"></a>
```

Le sélecteur :

```
div a { ... }
```

Ciblera les balises <a> qui sont dans la <div class=titre>

La balise <a> de l'accueil ne sera pas affectée.

Un sélecteur surchargé (**SubClasses**):

Exemple : CSS :

```
a :hover { ... }
```

Les balises <a> recevront le réglage quand la souris passe dessus.

Exemple : CSS :

```
p::first-letter { ... }
```

Le premier caractère de chaque balise <p> sera traité par le CSS

→ Il existe beaucoup d'autres combinaisons de sélecteurs (voir les sites de référence)

2. DES VARIABLES CSS

Très pratiques pour gérer globalement les couleurs d'un site.
On utilise le mot clé « var » et le double tiret.

Par exemple on définit au début du CSS :

```
:root {
  --couleur1: #6495ed;
  --couleur2: #faf0e6;
}
```

Ensuite, dans le fichier CSS on peut utiliser autant de fois qu'on veut la variable.

```
body { background-color: var(--couleur1); }
```

Il est donc très facile de changer les couleurs d'un site de façon globale en modifiant une seule valeur.

3. A DECOUVRIR ...

Le CSS (actuellement CSS3) s'améliore sans cesse.

A vous de vous documenter sur les extensions du CSS très performantes et qui parfois même permettent de se passer du JavaScript. Par exemple :

CSS-Animation	CSS Media Queries
CSS-Grid	CSS Flex Boxes

4. NOTATION BEM (BLOC – ELEMENT – MODULE)

C'est juste une façon de nommer les classes CSS.

Quand des blocs sont imbriqués et qu'ils ont chacun leur classe CSS, c'est une bonne idée d'utiliser la notation BEM comme dans cet exemple :

```
.project { } // Le bloc principal
.project__title {} // Un sous bloc (signes __)
.project__logo {} // Un autre sous bloc
.menu // Un au bloc principal
.menu__item {} // Un sous bloc
.menu__item--is-open {} // Une modification du sous bloc (signes --)
```

5. EVOLUTION SASS (SYNTACTICALLY AWESOME STYLE SHEET)

Un système très utilisé appelé SAAS permet de simplifier l'écriture du CSS. Le fichier sera de type SCSS.

La simplification consiste à supprimer les « ; », et utiliser les décalages (indentations) pour indiquer des éléments qui sont imbriqués dans d'autres. Les variables sont plus claires. C'est une alternative à BEM.

Inconvénient : Le navigateur ne sachant pas lire le SCSS, le fichier SCSS doit être converti en CSS avec un logiciel de conversion (Préprocesseur SASS) à installer dans votre environnement de développement.

Exemple :

```
$couleur1: #a2b9bc

header
  h1 // Balise h1 dans la balise <header>
    margin: 50px 0px 20px 0px
    text-align: center
    padding: 2rem
    background: $couleur1
    font-size: 40px

  p // Balise <p> dans la balise <header>
    font-size: 25px
    color: black
```