



WEB - Les bases du CSS

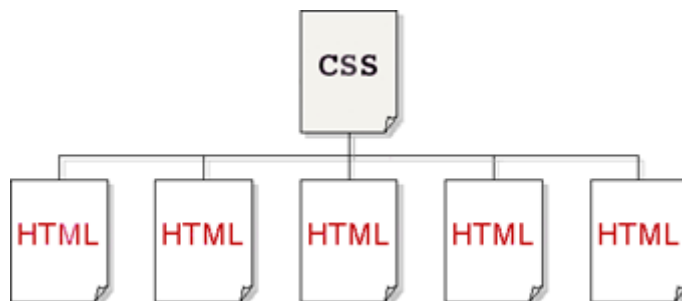
[Mise à jour le 25/8/2022]

- **Source**
 - [MDN web doc](#)

“CSS est l'un des langages principaux du Web ouvert et a été standardisé par le W3C. Ce standard évolue sous forme de niveaux (levels), CSS1 est désormais considéré comme obsolète, CSS2.1 correspond à la recommandation et CSS3, qui est découpé en modules plus petits, est en voie de standardisation. MDN Web Docs Mozilla”.

1. Les feuilles de style en cascade

« Le design d'un site évolue au fil du temps. Le **problème**, lorsqu'on n'utilise pas de feuilles de style, c'est qu'il faut reprendre toutes les pages HTML une à une pour modifier une police de caractère ou une couleur de fond... Avec les “**Cascading Style Sheets**” (**CSS**), ce lourd handicap est résolu. »



« **CSS** (Cascading Style Sheets : **feuilles de style en cascade**) est un **langage informatique** qui sert à décrire la **présentation** des documents HTML.

L'un des objectifs majeurs des CSS est de permettre la **mise en forme** hors des documents. Il est par exemple possible de ne décrire que la structure d'un document en HTML, et de décrire toute la **présentation dans une ou plusieurs feuilles de style CSS** séparées.

Ainsi, les avantages des feuilles de style sont multiples :

- La structure du document et la présentation peuvent être gérées dans des **fichiers séparés**.
- La conception d'un document se fait dans un premier temps **sans se soucier de la présentation**, ce qui permet d'être plus **efficace**.
- Dans le cas d'un site Web, la **présentation est uniformisée** : les documents (pages HTML) font référence aux mêmes feuilles de styles.

Cette caractéristique permet de plus une remise en forme rapide de l'aspect visuel.

Le **code HTML** est considérablement **réduit en taille et en complexité**, puisqu'il ne contient plus de balises ni d'attributs de présentation. » (Wikipédia)



2. Notion de sélecteur

Sélecteur : toute balise HTML privée des signes < et >. Le sélecteur “**accroche**” un **effet de style** à un élément ou une balise HTML. Chaque sélecteur (body, p, etc.) possède un ensemble de propriétés qui peuvent être définies par une ou plusieurs feuilles de style.

Syntaxe

```
selecteur {
  propriété1 : valeur1;
  propriété2 : valeur2;
  ...
  propriétén : valeurn;
}
```

Exemples

*.css

```
a {
  color: grey;
}

body {
  font-size: 12px;
  font-family: arial, sans-serif;
  background-color: rgba(8, 85, 157, 0.15);
}
```

```
}
```

A RETENIR

Le couple **propriété : valeur ;** constitue une **déclaration de style** et se termine par un point-virgule “;”.

Un **sélecteur** + une ou plusieurs **déclarations de style** constituent une **REGLE CSS**

3. Portée d'une feuille de style

3.1 Utilisation locale

La portée est **locale** si les **déclarations de style** sont placées **au sein de la balise** dans une propriété **style**.

Syntaxe

```
<balise style="propriété:valeur;">
...
</balise>
```

Exemple

*.html

```
<!DOCTYPE html>
<html>
<head>
...
</head>

<body style="background-color:blue">
...
</body>
</html>
```

3.2 Utilisation globale

La portée est **globale** lorsqu'elle a lieu dans l'**en-tête du document**. Elle est introduite par la balise **<style>**.

Exemple

*.html

```
<!DOCTYPE html>
<html>

<head>
  <style type="text/css">
    body {
      background-color: blue;
    }

    p {
      background-color: lightgreen;
      color: darkblue
    }

    h1 {
      font-family: helvetica;
    }
    ...
  </style>
</head>

<body>
  ...
</body>

</html>
```

3.3 Utilisation externe

Un lien vers une page .css est réalisé dans l'en-tête du document HTML avec une balise **<link>** et les propriétés **rel**, **type** et **href**.

- **Avantage** : application d'un ensemble de règles à toutes les pages d'un site.

Exemple

*.html

```
<!DOCTYPE html>
<html>
```

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

<body>
  ...
</body>
</html>
```

3.4 Priorité

PRIORITES

En cas de multiples définitions, **la définition locale est prioritaire sur la globale, elle-même prioritaire sur l'externe.**

4. Les cibles des principales propriétés

- Propriétés liées aux **polices de caractères**.
- Propriétés liées à la **mise en page des textes**.
- Propriétés liées aux **arrière-plans**.
- Propriétés liées aux **listes**.
- Propriétés liées aux **bordures**.
- Propriétés liées à la **définition des marges**.

5. Les classes de style

Les classes permettent d'appliquer **plusieurs styles différents** à une même balise. Le concepteur peut ainsi **définir ses propres sélecteurs**. Une classe est applicable à **plusieurs éléments**.

5.1 Les classes régulières

Elles permettent la définition de **plusieurs règles** d'affichage pour un **même sélecteur**.

- *Syntaxe*
 - **Déclaration** : *sélecteur.nom_classe* {déclaration(s) de style}
 - **Utilisation** : <balise **class**="nom_classe">...</balise>

- Exemple de déclaration

*.css

```
/* Déclaraion d'une classe gris qui sera appliquée à toutes les balises p */  
p {  
  color: grey;  
  font-style: italic;  
}
```

Ce texte est en gris et en italique.

- Exemple d'utilisation

*.html

```
<p>Ce texte est en gris et en italique.</p>
```

5.2 Les classes génériques

Elles ne sont pas liées à un sélecteur et peuvent être **appliquées à toutes les balises**.

- Syntaxe
 - **Déclaration** : `.nom_classe {déclaration(s) de style}`
 - **Utilisation** : `<balise class="nom_classe">...</balise>`
- Exemple de déclaration

*.css

```
/* Déclaraion d'une classe gris générique */  
.gris {  
  color: grey;  
  font-style: italic;  
}
```

Ce texte est en gris et en italique comme le précédent.

- Exemple d'utilisation

*.html

```
<!-- Il faut préciser le nom de la classe générique pour qu'elle  
s'applique à la balise -->  
<p class="gris">Ce texte est en gris et en italique.</p>
```

5.3 Les pseudo-classes

Elles autorisent des définitions pour l'affichage des balises dans des **états particuliers**.

- *Syntaxe*
 - **Déclaration** : *sélecteur:nom_pseudo_classe* {déclaration(s) de style}
- **Les pseudo-classes d'ancre : link, visited**
 - *Exemple* : couleur du fond d'un lien déjà visité rouge et la couleur du texte associé en blanc.

*.css

```
a:visited{  
  color:white;  
  background-color: red;  
}
```

- **Les pseudo-classes dynamiques : hover, active, focus**
 - *Exemple* : mise en italique du lien au passage de la souris.

*.css

```
a:hover{  
  font-style:italic;  
}
```

6. Le sélecteur id

Ce sélecteur accroche un effet de style (comme le sélecteur class) mais **ne peut être appelé qu'une seule fois** dans le document. Il permet d'identifier un élément unique dans la page (**très utile lorsqu'on fait appel à du code JavaScript**).

- *Syntaxe*
 - **Déclaration** : **#nom_classe** {déclaration(s) de style}
 - **Utilisation (appel une seule fois)** : **<balise id="nom_classe">...</balise>**

- Exemple

*.html

```
<!DOCTYPE html>
<html>

<head>
  <style type="text/css">
    #tcan {
      color: blue;
      font-size: 20px;
    }
  </style>
</head>

<body>
  <h1 id="tcan">La conversion analogique numérique</h1>
  <p> Un convertisseur analogique-numérique est un dispositif
  électronique ...</p>
  ...
</body>

</html>
```

La conversion analogique numérique

Un convertisseur analogique-numérique est un dispositif électronique ...

? Les balises de type bloc et en ligne

- HTML

La plupart des balises peuvent se ranger dans l'une ou l'autre des **catégories** : **bloc** ou **en ligne**.

Une balise de type **bloc** crée automatiquement un **retour à la ligne** avant et après son contenu. Une page web est constituée d'un ensemble de blocs que l'on peut mettre les **uns à l'intérieur des autres**.

Exemples

*.html

```
<h1></h1>
<p></p>
<div></div>
```


Une balise de type **en ligne** se trouvera **obligatoirement à l'intérieur** d'une balise de type bloc. Une balise de type en ligne ne crée pas de retour à la ligne.

Exemples

*.html

```
<a></a>
<em></em>
<span></span>
...
```

- **CSS**

La propriété CSS **display** permet de transformer n'importe quel élément d'un type dans un autre. Cette propriété prend les valeurs ci-dessous.

Valeur	Description
inline	Les éléments se placent les uns à côté des autres.
block	Les éléments sont en forme de blocs. Ils se placent les uns en dessous des autres et peuvent être redimensionnés.
inline-block	Les éléments sont positionnés les uns à côté des autres (comme inline) et peuvent être redimensionnés (comme block).
none	Les éléments ne seront pas affichés.

Exemple : Positionnement de liens les uns en dessous des autres et redimensionnement possible.

*.css

```
a {
  display : block ;
}
```

From:

<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<https://webge.fr/dokuwiki/doku.php?id=web:basescss&rev=1661422972>

Last update: **2022/08/25 12:22**

