

CSS

INTRODUCTION

La CSS est le langage qui définit l'apparence et l'identité visuelle d'une page web. On l'appelle aussi « feuille de style ». Elle est complémentaire au langage HTML et permet de modifier la couleur, la taille, le positionnement, etc. des différents éléments.

En savoir plus

CSS signifie Cascading Style Sheets.

Ce langage est inventé en 1996, soit 5 ans après l'invention de l'HTML. Tout comme le langage HTML, la CSS évolue continuellement.

La CSS 3 est la dernière version, elle donne de nombreuses possibilités esthétiques et permet également de faire des animations.

LIER SA FEUILLE DE STYLE CSS

Comme pour le fichier HTML la CSS se crée depuis un éditeur de texte en l'enregistrant en **.css**. Pour associer une feuille de style à son fichier HTML il faut placer une balise `<link>` dans le `<head>` de notre fichier HTML. On va lui donner les attributs pour lui indiquer de quel type de ressource il s'agit `rel='stylesheet'` et `type='text/css'`. Ainsi que son « chemin » pour lui dire où elle se trouve par rapport à notre fichier HTML `href='css/style.css'`.

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

On peut également mettre de la CSS dans la partie `<head>` à l'intérieur d'une balise `<style>`.

```
<head>
  <style>
    /* ici on peut mettre de la CSS */
  </style>
</head>
```

Ou encore dans un attribut directement sur la balise, comme ceci :

```
<p style="/* ma CSS */">
```

À noter : Un commentaire en CSS s'écrit sous cette forme :

```
/* Ceci est un commentaire*/
```

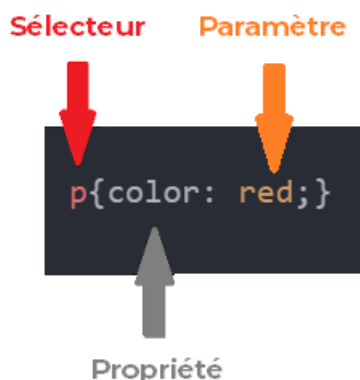
Je vous conseil fortement de lier une feuille de style, c'est la meilleur méthode pour écrire de la CSS. En effet la balise `<style>` va allonger votre fichier HTML et donc le rendre moins lisible. Cette méthode peut être utile dans une petite page web. L'attribut de `style` va quant à lui avoir un problème de réutilisation, lors d'une modification ou pour donner un même style sur plusieurs éléments. Je vous déconseille fortement cette dernière méthode.

MODIFIER LES PROPRIÉTÉS

Maintenant que notre feuille de style est liée nous allons pouvoir modifier l'apparence des balises. Il existe un grand nombre de propriété que l'on peut modifier.

Couleur du texte

Par exemple, on va modifier la couleur des balises de paragraphe `<p>` pour les afficher en rouge. Pour cela il faut donc **sélectionner** la balise à modifier. Ensuite, on indique que c'est **la propriété** de la couleur que l'on souhaite changer. Puis on lui donne son nouveau **paramètre**. On écrit alors :



« `p` » est donc la balise `<p>` que l'on souhaite modifier. Les accolades servent à déclarer les propriétés et leurs paramètres.

« `color` » est le nom de la propriété que l'on va changer.

« `red` » est le nouveau paramètre de la propriété.

Le point-virgule est indispensable, il signifie qu'on a terminé de changer la propriété.

À noter : Si vous oubliez de finir par le point-virgule la modification ne sera pas prise en compte. La syntaxe est très importante.

Taille du texte

Pour agrandir ou réduire la taille du texte on utilise la propriété `font-size` puis on indique une valeur.

```
p{font-size: 20px;}
```

« `px` » signifie **pixel**, c'est son unité.

On peut modifier plusieurs propriétés, on écrit alors :

```
p{color: red; font-size : 20px ;}
```

Lorsque l'on a beaucoup de propriétés, on va préférer une écriture sur plusieurs lignes pour plus de clarté :

```
p{
  color: red;
  font-size : 20px ;
}
```

Le fond

On peut aussi changer la couleur du fond avec la propriété `background-color` par exemple :

```
body{background-color: silver;}
```

Il est possible de mettre une image de fond avec la propriété `background-image` par exemple :

```
body{background-image: url(images/mon-image.jpg);}
```

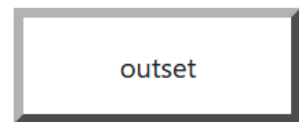
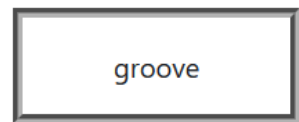
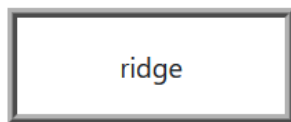
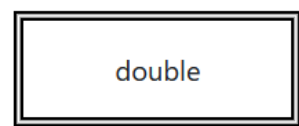
« `url` » est l'équivalent de l'attribut `src` sur la balise `` en HTML. Il donne son chemin qui est composé de son emplacement par rapport au fichier, du nom de l'image et de son extension (en général `.jpg` ou `.png`).

Les bordures

Pour ajouter des bordures on utilise la propriété `border`. On doit lui donner plusieurs paramètres : sa taille, son type et sa couleur. On écrit alors :

```
p{border: 2px solid red;}
```

Il existe plusieurs types de bordure :



On peut aussi arrondir la bordure en ajoutant la propriété `border-radius`.

```
p{border : 2px solid red; border-radius : 10px;}
```

MISE EN PAGE

Les marges

Les propriétés `margin` et `padding` servent toutes les deux à ajouter des marges aux éléments, mais elles ont une grande différence qu'il faut comprendre.

- `margin` : ajoute les marges sur l'extérieur de l'élément.
- `padding` : ajoute les marges à l'intérieur de l'élément.

On peut mettre des marges différentes en fonction des côtés. Si on souhaite mettre une marge sur un seul côté on utilise alors la propriété approprié :

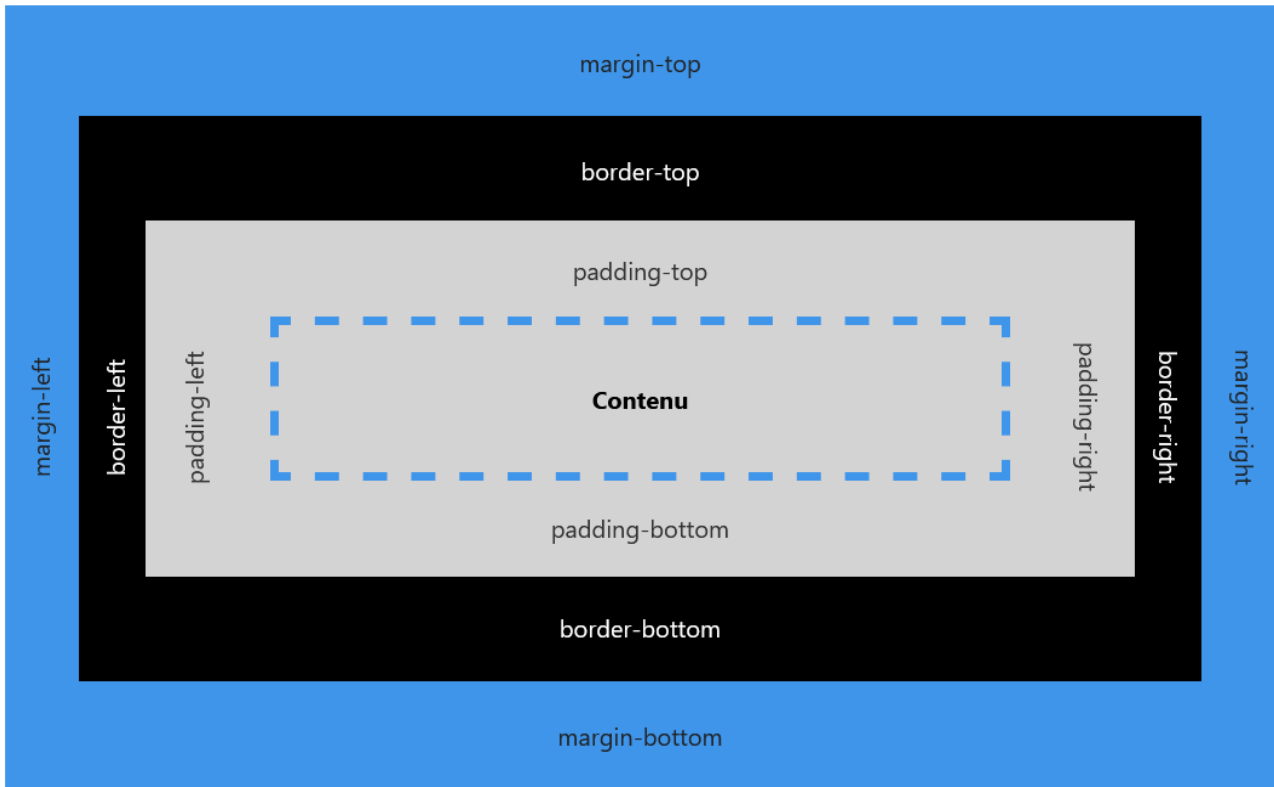
- `margin-top` : ajoute une marge extérieure en haut.
- `margin-right` : ajoute une marge extérieure à droite.

- `margin-bottom` : ajoute une marge extérieure en bas.
- `margin-left` : ajoute une marge extérieure à gauche.

Comme pour les `padding` :

- `padding-top` : ajoute une marge intérieure en haut.
- `padding-right` : ajoute une marge intérieure à droite.
- `padding-bottom` : ajoute une marge intérieure en bas.
- `padding-left` : ajoute une marge intérieure à gauche.

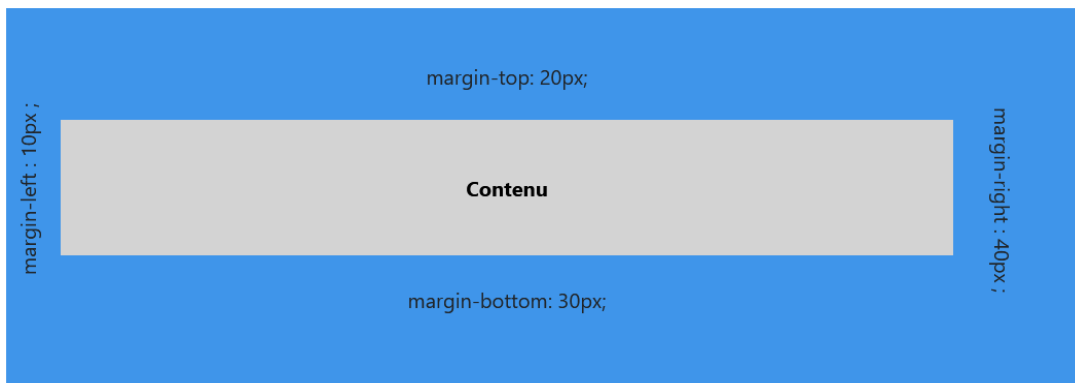
Voici un exemple plus visuel :



Si on veut mettre une marge différente en fonction des côtés on peut alors écrire :

```
margin: 20px 40px 30px 10px;
```

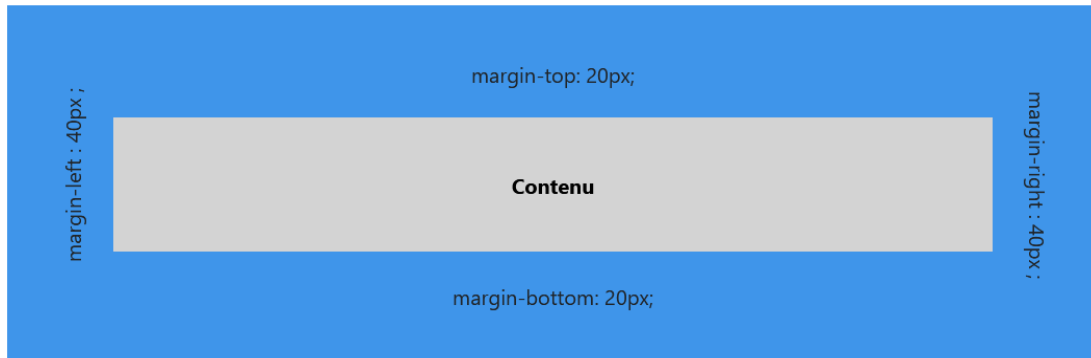
Les paramètres vont s'ajouter en commençant par le haut puis en tournant dans le sens horaire. Qui correspond aux valeurs dans l'ordre : top, right, bottom et left. On aura donc :



Si on donne que deux valeurs :

```
margin: 20px 40px;
```

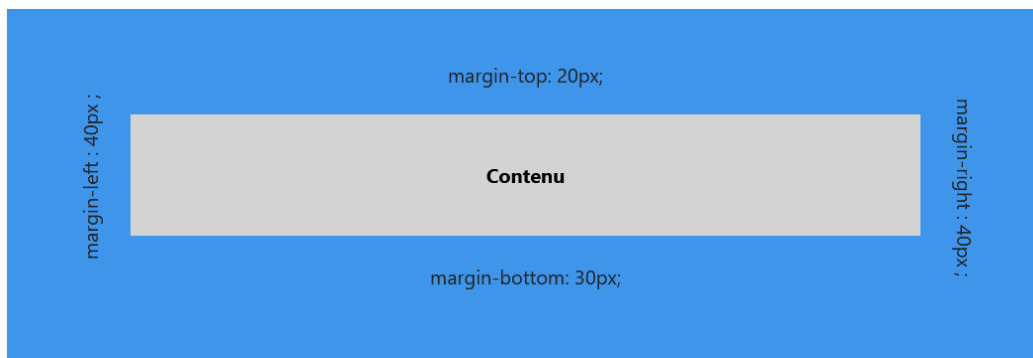
Alors, elles se répète comme si on avait écrit `margin : 20px 40px 20px 40px ;` ; On aura donc **20px** de marges en haut comme en bas et **40px** sur le côté droit comme le côté gauche.



Si on donne trois valeurs :

```
margin: 20px 40px 30px;
```

Alors, la valeur manquante (celle du côté gauche) va se compléter comme par miroir en répétant la valeur du côté droit. Comme si on avait écrit `margin : 20px 40px 30px 40px ;` ; On aura donc **20px** de marges en haut, **30px** en bas, **40px** sur le côté droit et sur la gauche.



Largeur et Hauteur d'un élément

On peut définir la largeur et la hauteur d'un élément par la propriété `width` pour la largeur et `height` pour la hauteur. Il suffit ensuite de lui indiquer la valeur.

```
p{  
  width: 200px; /* donne une largeur de 200 pixels */  
  height: 300px ; /* donne une hauteur de 300 pixels */  
}
```

LES IDENTIFIANTS ET LES CLASSES

Pour modifier un élément précis on peut lui attribuer **un identifiant**. Pour cela du côté HTML on va lui ajouter un attribut `id` et de lui indiquer un nom.



La balise à maintenant un identifiant que l'on peut cibler. Côté CSS il suffit de sélectionner cet identifiant, on écrit alors :

```
#demo{color: red;}
```

Le dièse signifie que l'on cible un attribut de type `id`, un identifiant. Suivi de son nom, ici nommé « demo ».

Attention : un identifiant est unique, deux éléments ne peuvent pas avoir le même identifiant.

Pour modifier plusieurs éléments on utilise alors **une classe**. Contrairement aux identifiants celles-ci peuvent se répéter sur plusieurs éléments. Pour cela du côté HTML on va lui ajouter un attribut `class` et de lui indiquer un nom.



La balise à maintenant une classe que l'on peut cibler. Côté CSS il suffit de sélectionner cette classe, on écrit alors :

```
.demo{color: red;}
```

Le point signifie que l'on cible une classe, suivi de son nom.

À noter : Les noms des identifiants et des classes doivent s'écrire en minuscule et ne peuvent pas commencer par un chiffre.

IMPORTANTANCE ET HÉRITAGE DES SÉLECTEURS CSS

En CSS les éléments enfants héritent naturellement de certaines propriétés des éléments parents. Il y a aussi une notion d'importance.

Pour l'héritage, prenons l'exemple suivant :

HTML

```
<div>
  <p>Texte de paragraphe.</p>
  <p><span>Autre paragraphe.</span></p>
</div>
```

CSS :

```
div{color: blue;}
```

Rendu :

Texte de paragraphe.

Autre paragraphe.

Les `<p>` enfants de `<div>` héritent de la propriété, tout comme le `` qui est l'enfant du `<p>`.

Pour l'importance des sélecteurs, prenons l'exemple suivant :

HTML

```
<p>Texte de paragraphe.</p>
<p class="texte-vert">Texte écrit en vert.</p>
<p id="texte-bleu" class="texte-vert">Texte écrit en bleu.</p>
```

CSS

```
p{color:gray;}
.texte-vert{color:green;}
#texte-bleu{color:blue;}
```

Rendu :

Texte de paragraphe.

Texte écrit en vert.

Texte écrit en bleu.

L'identifiant à une importance plus haute qu'une classe et une classe à une importance plus haute qu'une balise. On augmente l'importance en ajoutant des sélecteurs, pour cibler plus précisément.

Exemple, sur les ajout de sélecteurs :

HTML

```
<p>Texte de paragraphe.</p>
<div>
  <p>Texte écrit en gris.</p>
  <p class="texte-bleu texte-jaune">Texte écrit en vert.</p>
  <p id="texte-bleu">Texte écrit en bleu.</p>
</div>
```

CSS

```
p{color: purple;}
div p{color: gray;}
.texte-bleu.texte-jaune{color: green;}
#texte-bleu, .texte-bleu{color: blue;}
```

Rendu :

Texte de paragraphe.

Texte écrit en gris.

Texte écrit en vert.

Texte écrit en bleu.

- L'espace entre deux sélecteurs permet de cibler les enfants (ici on cible `<p>` l'enfant de `<div>`).
- On peut ajouter plusieurs classes sur un élément. Si on colle les deux sélecteurs de classe, alors on cible seulement les éléments qui auront ces deux classes (ici on cible l'élément qui a la classe « texte-bleu » et qui a aussi la classe « texte-jaune »).
- La virgule permet d'ajouter un sélecteur qui aura la même propriété que le précédent (ici l'`id` « texte-bleu » tout comme la classe « texte-bleu » auront la même couleur).

Il existe d'autres sélecteurs CSS. Je vous invite à consulter la documentation pour en prendre connaissance.