



LES NOUVEAUTES DE CSS3

1.	Les sélecteurs	1
1.1.	Les sélecteurs d'attributs	2
1.2.	Le sélecteur d'adjacence directe	3
1.3.	Les pseudo-classes	4
1.4.	Les pseudo-éléments CSS3	6
2.	Les bordures	7
3.	Les backgrounds	9

1. Les sélecteurs

Dans la séquence 1 portant sur les bases du CSS, vous avez appris l'utilité des sélecteurs. Pour rappel, un sélecteur permet tout simplement d'affecter un style à un élément HTML.

Exemples :

- Dans cet exemple, le sélecteur **P** cible tous les paragraphes de notre page html.

```
P
{
    color: green;
}
```

- Appliquer un style à tous les titres

```
h1
{
```

```
}
```

- Utiliser class et id

```
.class
{

}
#id
{

}
```

Les nouveaux sélecteurs de CSS3 permettent d'atteindre des nœuds du document HTML de manière plus précise. Les nouveautés incluent :

- les sélecteurs d'attributs,

- le sélecteur d'adjacence directe ;
- les pseudo-classes ;
- les pseudo-éléments.

1.1. Les sélecteurs d'attributs

La norme CSS3 a introduit 3 nouveaux sélecteurs d'attributs qui se différencient par la position de la chaîne de caractères recherchée dans la valeur de l'attribut :

- E[A^="C"] : permet de sélectionner un élément E dont la valeur de l'attribut A commence par la chaîne de caractère C.

Exemple :

Code CSS

```
p.example{
    margin:0;
    padding:10px;
    color:#000;
}
p.example[title^="ess"]{
    color:#fff;
    background:#333;
}
```

Code HTML

```
<p class="example"> je n'ai pas d'attribut title</p>
<p class="example" title="comment"> j'ai un attribut title mais il ne commence pas par "ess" </p>
<p class="example" title="essai"> j'ai un attribut title commençant par "ess" </p>
<p class="example" title="esson"> j'ai un attribut title commençant par "ess" également</p>
```

- E[A\$="C"] : permet de sélectionner un élément E dont la valeur de l'attribut A se termine par la chaîne de caractère C.

Exemple :

Code CSS

```
p.example2{
    margin:0;
    padding:10px;
    color:#000;
}
p.example2[title$="sai"]{
    color:#fff;
    background:#045FB4;
}
```

Code HTML

```
<p class="example2"> je n'ai pas d'attribut title</p>
<p class="example2" title="comment"> j'ai un attribut title mais il ne commence pas par "ess" </p>
<p class="example2" title="essai"> j'ai un attribut title commençant par "ess" </p>
<p class="example2" title="esson"> j'ai un attribut title commençant par "ess" également</p>
```

- E[A*="C"] : permet de sélectionner un élément E dont la valeur de l'attribut A contient par la chaîne de caractère C.

Exemple :

Code CSS

```
p.example3{
```

```

        margin:0;
        padding:10px;
        color:#000;
    }
    p.example3[title*="val"]{
        color:#fff;
        background:#990000;
    }

```

Code HTML

```

<p class="example3"> je n'ai pas d'attribut title</p>
<p class="example3" title="comment"> j'ai un attribut title mais il ne contient pas "val" </p>
<p class="example3" title="val"> j'ai un attribut title contenant au moins "val" </p>
<p class="example3" title="evaluer"> j'ai un attribut title contenant au moins "val" également</p>
<p class="example3" title="eval"> j'ai un attribut title contenant au moins "val" également</p>

```

Les exemples précédents permettent de sélectionner un seul attribut. Par exemple

```

a[href$=".doc"]{
background :rgb(0,255,255) ;
}

```

permet de sélectionner les liens qui se terminent par .doc

Il est possible de faire une sélection d'attributs multiples :

Exemple :

HTML

```

<article>
<p>
<a href="document.doc"> lien document interne </a>
<br />
<a href="http://www.site.fr/document.doc"> lien document externe </a>
</p>
</article>

```

CSS

```

a[href$=".doc"]{
background :rgb(0,255,255) ;
}
a[href^=".http"][href$=".doc"]{ /* l'attribut débute doit débiter par http */
background :rgb(255,255,0) ; /* et doit finir par .doc */
}

```

1.2. Le sélecteur d'adjacence directe

Toujours dans le souci d'atteindre des nœuds du document HTML de manière plus précise, vous pouvez utiliser un **sélecteur d'adjacence indirecte**. Ce dernier permet d'ajouter un style à tous les éléments qui suivent un élément particulier.

Voici la syntaxe :

E ~ F : permet de cibler les éléments F précédés par l'élément E.

Exemple :

Code CSS

```

.example4 div{
    margin:0;
    padding:10px;
    color:#000;
}
.example4 div~p{
    color:#fff;
    margin:20px;
    width:200px;
    padding:5px;
}

```

```

border:1px solid #333;
background:#006644;
}

```

Code HTML

```

<div class="example4">

    <div>je suis l'élément particulier div</div>
    <p> je suis un p qui suit le div (l'élément particulier)</p>
    <p>je suis un p qui suit le div (l'élément particulier)</p>
    <span>je suis un span</p>
    <p>je suis un p qui ne suit pas le div (l'élément particulier)</p>

</div>

```

1.3. Les pseudo-classes

Parmi un ensemble d'éléments possédant le même parent, comme les éléments d'une liste ou les lignes d'un tableau, il est fastidieux en CSS2 de cibler un élément particulier par sa position. Pour remédier à cette difficulté, les CSS3 introduisent le ciblage d'un énième enfant d'une balise parente.

➤ P E:nth-child(position)

Il permet de cibler l'énème enfant E de l'élément parent P, en débutant le comptage de la position par le premier enfant, et en descendant dans la structure HTML.

Exemple :

CSS

```

.exampleTable{
    width:100%;
    border:1px solid #444;
}
.exampleTable tr:nth-child(even){ //tous les enfants aux numéros pairs
    background:#999999;
    text-shadow: 2px 2px 5px #111;
    color:#fff;
}
.exampleTable tr:nth-child(odd){ //tous les enfants aux numéros impairs
    background:#990000;
    color:#fff;
}
.exampleTable tr:nth-child(3n){ //tous les 3 enfants
    background:#045FB4;
    color:#fff;
}
.exampleTable tr:nth-child(7){ //l'enfant numéro 7
    background:#006400;
    text-shadow: 2px 2px 2px #fff;
    color:#000;
}
}

```

HTML

```

<table class="exampleTable">
    <tr>
        <td>1ere ligne</td>
    </tr>
    <tr>

```

```

        <td>2eme ligne</td>
    </tr>
    <tr>
        <td>3eme ligne</td>
    </tr>
    <tr>
        <td>4eme ligne</td>
    </tr>
    <tr>
        <td>5ere ligne</td>
    </tr>
    <tr>
        <td>6eme ligne</td>
    </tr>
    <tr>
        <td>7eme ligne</td>
    </tr>
    <tr>
        <td>8eme ligne</td>
    </tr>
</table>

```

➤ P E:nth-last-child(position)

Ce sélecteur accepte les mêmes arguments que :nth-child() et correspond au dernier enfant d'un élément parent. C'est le même principe que le :nth-child() sauf que l'on part de la fin.

Exemple :

CSS

```

.exampleTable2{
    width:100%;
    border:1px solid #444;
}
.exampleTable2 tr:nth-last-child(odd){ /*tous les enfants aux numéros impairs depuis la fin.*/
    background:#990000;
    color:#fff;
}
.exampleTable2 tr:nth-last-child(-n+2){ /*les 2 derniers enfants.*/
    background:#045FB4;
    color:#fff;
}
tableau.*.exampleTable2 tr:nth-last-child(7){ /*l'enfant numéro 7 en partant de la fin donc la 2ème ligne du
    background:#006400;
    text-shadow: 2px 2px 2px #fff;
    color:#000;
}

```

HTML

```

<table class="exampleTable2">
    <tr>
        <td>1ere ligne</td>
    </tr>
    <tr>
        <td>2eme ligne</td>
    </tr>
    <tr>
        <td>3eme ligne</td>
    </tr>

```

```

<tr>
    <td>4eme ligne</td>
</tr>
<tr>
    <td>5ere ligne</td>
</tr>
<tr>
    <td>6eme ligne</td>
</tr>
<tr>
    <td>7eme ligne</td>
</tr>
<tr>
    <td>8eme ligne</td>
</tr>
</table>

```

➤ **P E:nth-of-type(position)**

Il permet de cibler l'énème occurrence du type de l'élément E présent dans l'élément parent P, en débutant le comptage de la position par le premier enfant de type E, et en descendant dans la structure HTML.

Exemple :

```

img:nth-of-type(2n+1) { float: right; }
img:nth-of-type(2n) { float: left; }

```

➤ **P E:nth-last-of-type (position)**

Il permet de cibler l'énème occurrence du type de l'élément E présent dans l'élément parent P, en débutant le comptage de la position par le dernier enfant de type E, et en remontant dans la structure HTML.

Exemple :

```

body > h2:nth-of-type(n+2):nth-last-of-type(n+2)

```

La position de l'élément ciblé se spécifie avec un paramètre qui peut être de trois types différents :

- **un nombre entier positif** : représente une position unique dont le comptage a débuté à 1.
- **un multiplicateur** : représente des multiples pouvant contenir une opération arithmétique.
- **un mot clé** : représente les positions paires ou impaires.

1.4. Les pseudo-éléments CSS3

En CSS2, les pseudo-éléments **first-line**, **first-letter**, **before**, **after** étaient précédés d'un unique caractère « deux-points ». Désormais, la syntaxe officielle nécessite deux caractères « deux-points ».

L'ancienne syntaxe reste rétro compatible.

::first-line : applique la règle de style à la première ligne du texte de l'élément.

Exemple :

```

p::first-line { text-transform: uppercase } /* La 1ère ligne de ou des éléments "p" est mise en majuscule.*/

```

::first-letter: applique la règle de style à la première lettre du texte de l'élément.

Exemple :

`p::first-line { font-size: 2em } /* La 1ère ligne de ou des éléments "p" est mise en majuscule.*/`

::selection: applique la règle de style à la sélection du texte de l'élément faite par l'utilisateur.

Exemple :

`p::selection { background:#006644 } /* La 1ère ligne de ou des éléments "p" est mise en majuscule.*/`

::before et :: after : génère un contenu avant et après un contenu d'élément.

2. Les bordures

Le CSS3 améliore la propriété border afin de lui attribuer plus de fonctionnalités.

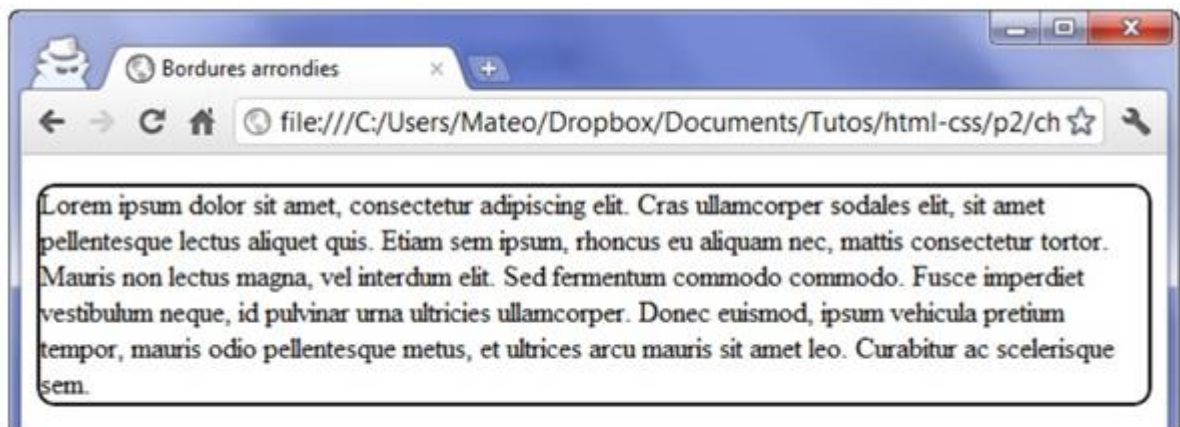
➤ Border-radius

Avec cette propriété, vous pouvez arrondir les angles de la bordure. Cependant, il est nécessaire de prendre en compte le navigateur que vous utilisez en ajoutant un préfixe (-webkit, -moz, -o).

Exemple :

```
p
{
    border-radius : 10px ;
}
```

Voici le résultat :



Remarque : Si les 4 valeurs des coins (haut, droit, bas, gauche) sont égales, il suffit d'indiquer une seule des valeurs.

`border-radius : 10px` est équivalent à `border-radius : 10px 10px 10px 10px ;`

Exemple :

```
border-radius: 10px 0px 0px 0px;
```

➤ **Border-color**

Cette propriété permet d'attribuer à la bordure.

Exemple :

```
border-color: #900 #064 #990 #960;
```

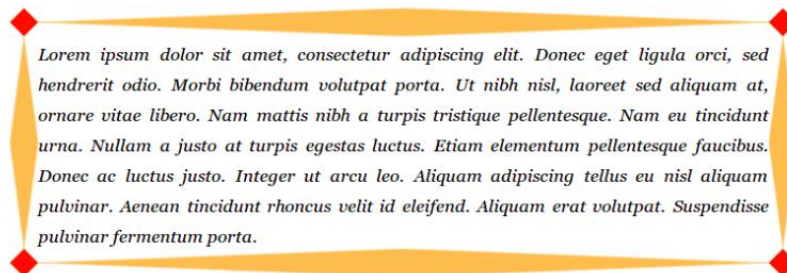
➤ **Border-image**

La bordure peut être dorénavant mise en forme via des **images**. On peut ainsi spécifier une ou des images pour chaque bordure (top, left, bottom, right) ainsi qu'une ou des images pour les coins des bordures via les propriétés suivantes :

- border-image;
- border-top-image ;
- border-right-image ;
- border-bottom-image ;
- border-left-image.

Exemple :

```
div {  
  border: double orange 1px;  
  border-image-source: url(small-border.png);  
  border-image-slice: 27;  
}
```

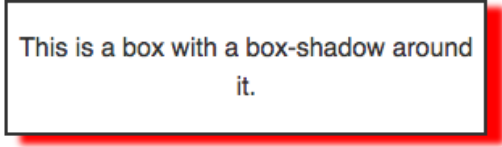


➤ **Box-shadow**

Cette propriété permet de générer une ombre portée sur n'importe quel élément HTML.

Exemple :

```
box-shadow: 10px 5px 5px red;
```

This is a box with a box-shadow around it.

3. Les backgrounds

CSS3 a introduit de nouvelles propriétés permettant de contrôler l'apparence du fond d'un élément.

➤ Background-size

Il permet de définir la taille de l'image de fond.

Syntaxe : `background-size : width height (cover/contain) ;`

cover : l'image de fond couvre toute la boîte

contain : l'image de fond est contenue dans la surface définie mais sans déformation.

D'autres options d'utilisation de cette propriété sont disponibles dans ce lien :

<https://www.alsacreations.com/tuto/lire/1390-arriere-plans-css3-background.html>

➤ Background-clip

Il permet de définir les limites des backgrounds suivants trois variantes content, padding et border.

Syntaxe : `background-clip : [border|padding|content] ;`

border : le fond s'étend jusqu'aux bordures

padding : le fond se limite aux marges extrêmes.

content : le fond reste dans la limite du contenu.

Exemple : <https://www.alsacreations.com/tuto/lire/816-css3-background-clip.html>


➤ Background dégradés

Deux sortes de dégradés sont disponibles : linéaires ou radiaux.

Exemple :

```
background: linear-gradient(to right, red, orange, yellow, green, blue, indigo, violet);
```

CSS



Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Source : <https://www.alsacreations.com/tuto/lire/813-css3-background-radial-linear-gradient.html>

➤ Background multiples

Cette propriété permet d'afficher plusieurs images en arrière plan.

Voir : <https://www.alsacreations.com/tuto/lire/814-css3-background-image-multiple.html>