

# Savoir comment utiliser les sélecteurs CSS 2.1 et les nouveautés CSS 3

par Julien ([Kiwano](#), [Référencement Marseille](#))

Date de publication : 07/07/2009

Les sélecteurs CSS sont certainement un des aspects les plus puissants en CSS. Ils permettent de cibler un élément spécifique du balisage HTML pour lui affecter un style. Les sélecteurs CSS sont variés, allant du simple nom d'un élément jusqu'à une expression de sélection riche d'une liste d'éléments HTML. Cet article vous fait un rappel sur les possibilités des sélecteurs CSS et vous présente les nouveautés CSS 3 qui s'avèrent très puissantes pour éviter des *id*, classes ou JavaScript non nécessaires.

Sélecteur CSS 2.1.....	3
I-A - Syntaxe des sélecteurs.....	3
I-B - Sélecteur universel.....	3
I-C - Sélecteur de type.....	3
I-D - Sélecteur descendant.....	3
I-E - Sélecteur d'enfant.....	3
I-F - Sélecteur d'enfant adjacent.....	4
I-G - Sélecteur d'attribut.....	4
I-G-1 - [att].....	4
I-G-2 - [att=val].....	5
I-G-3 - [att~=val].....	5
I-G-4 - [att =val].....	5
I-H - Sélecteur de classe.....	5
I-I - Sélecteur d'ID.....	6
I-J - Pseudo-éléments et pseudo-classes.....	6
I-J-1 - La pseudo-classe :first-child.....	6
I-J-2 - Les pseudo-classes d'ancre :link et :visited.....	6
I-J-3 - Les pseudo-classes dynamiques :hover, :active et :focus.....	6
I-J-4 - La pseudo-classe de langue :lang.....	6
I-J-5 - Le pseudo-élément :first-line.....	7
I-J-6 - Le pseudo-élément :first-letter.....	7
I-J-7 - Les pseudo-éléments :before et :after.....	7
Nouveautés CSS 3 sur les sélecteurs.....	7
II-A - Sélecteur d'attribut.....	7
II-B - Combinateur d'adjacence indirecte.....	7
II-C - Pseudo-classes.....	7
II-C-1 - La pseudo-classe :nth-child().....	8
II-C-2 - La pseudo-classe :nth-last-child().....	8
II-C-3 - La pseudo-classe :last-child.....	8
II-C-4 - La pseudo-classe :checked.....	8
II-C-5 - La pseudo-classe :empty.....	8
II-C-6 - La pseudo-classe de negation.....	8
II-D - Pseudo-élément.....	9
II-D-1 - Les pseudo-éléments fragments d'éléments d'interface.....	9
Pour en savoir plus.....	9

## Sélecteur CSS 2.1

### I-A - Syntaxe des sélecteurs

Les sélecteurs peuvent être de type simple (cas du sélecteur universel), de type sélecteurs d'attributs ou encore de type sélecteurs d'id.

Les sélecteurs de type simple peuvent être combinés à l'aide de caractères spéciaux comme un espace, un "+" ou un ">".

Enfin on peut combiner les sélecteurs dans une liste quand on veut appliquer un même style aux éléments ciblés par plusieurs sélecteurs.

```
h1 { color: red; }
h2 { color: red; }
h1, h2 { color: red; } /* équivalent aux 2 lignes ci-dessus */
```

### I-B - Sélecteur universel

Le sélecteur universel correspond au caractère "\*", il représente l'ensemble des éléments de l'arbre HTML et peut être omis.

```
*.warning
.warning
/* les 2 syntaxes sont équivalentes */
```

### I-C - Sélecteur de type

Un sélecteur de type correspond au nom de l'élément HTML. Il permet d'appliquer un style à tous les éléments de ce type dans le document HTML. La règle suivante concerne tous les éléments *h1*.

```
h1 { font-family: sans-serif; }
```

### I-D - Sélecteur descendant

Un sélecteur descendant permet de cibler les éléments enfants d'un autre élément.

L'exemple suivant appliquera une couleur jaune au texte de tous les éléments *strong* contenus dans un paragraphe.

```
p strong { color: yellow; }
```

### I-E - Sélecteur d'enfant

Un sélecteur d'enfant est utilisé pour vérifier qu'un élément est un enfant d'un autre élément. Pour utiliser le sélecteur d'enfant on utilise le caractère ">" avec à gauche le parent et à droite l'enfant.

L'exemple suivant appliquera un fond bleu aux éléments *strong* enfants d'un paragraphe.

```
p > strong { background-color: blue; }
```

La différence entre le sélecteur d'enfant et le sélecteur descendant est que le sélecteur descendant correspond aux éléments *strong* où qu'ils soient dans le paragraphe, alors que le sélecteur d'enfant est plus spécifique et correspond uniquement aux éléments *strong* qui sont enfants du paragraphe.

Dans l'exemple suivant :

```
<p>Martin dit :  
  <q>super site sur le référencement naturel à <strong>Marseille</strong></q>,  
  en plus il connaissait bien les sujets <strong>SEO</strong>!  
</p>
```

Le code sélecteur descendant précédent mettra en jaune les deux textes entre les balises *strong* alors que le code sélecteur d'enfant précédent appliquera un fond bleu uniquement au 2e *strong* soit le mot SEO car seulement cet élément est un enfant du paragraphe. L'élément *strong* Marseille est un enfant de la citation *q*.

 Les sélecteurs d'enfants ne sont pas supportés par IE6.

## I-F - Sélecteur d'enfant adjacent

Les enfants adjacents sont les éléments qui suivent immédiatement un autre élément qui partage le même parent. On utilise le signe "+" pour les enfants adjacents.

L'exemple suivant appliquera un fond vert à tous les éléments *h2* qui suivent immédiatement un élément *h1* si ses éléments partagent le même parent.

```
h1 + h2 { background-color: green; }
```

 Les sélecteurs d'enfants adjacents ne sont pas supportés par IE6.

## I-G - Sélecteur d'attribut

Les sélecteurs d'attribut correspondent aux éléments qui ont des attributs définis dans l'arbre HTML. Il y a 4 types de sélecteurs d'attribut.

### I-G-1 - [att]

Le sélecteur d'attribut basique correspond aux éléments qui ont l'attribut spécifié indépendamment de la valeur de l'attribut.

L'exemple suivant appliquera un fond rouge à tous les liens qui possèdent un attribut *title*.

```
a[title] { background-color: red; }
```

## I-G-2 - [att=val]

Ce sélecteur d'attribut est utilisé pour sélectionner les éléments qui contiennent un attribut spécifique avec une valeur particulière.

L'exemple suivant appliquera un fond rouge à tous les liens qui possèdent un attribut *rel* avec la valeur *external*.

```
a[rel=external] { background-color: red; }
```

## I-G-3 - [att~=val]

Certains attributs autorisent une liste de valeurs séparées par des espaces. En utilisant le "~" avant le caractère "=", vous pouvez positionner un style sur les éléments qui possèdent un attribut spécifique avec une valeur particulière présente dans la liste des valeurs.

L'exemple suivant appliquera un fond rouge à tous les paragraphes qui contiennent la classe *attention*.

```
p[class~=marseille] { background-color: red; }  
/* p[class~=marseille] est équivalent à p.marseille */
```

## I-G-4 - [att|=val]

Ce sélecteur d'attribut est utilisé pour sélectionner les éléments qui possèdent un attribut spécifique et dont la valeur est une liste de mots séparés par des tirets et commençant par *val*.

L'exemple suivant appliquera le style sur les liens avec un *hreflang* dont la valeur commence par "en", incluant "en-US", "en-GB", etc.

```
a[hreflang|=en] { background-color: red; }
```

Plusieurs sélecteurs d'attributs peuvent être utilisés dans la même instruction.

L'exemple suivant utilise 2 sélecteurs d'attribut pour appliquer un fond rouge à tous les liens qui ont un attribut *title* et dont l'attribut *rel* à la valeur *external*.

```
a[title][rel~=external] { background-color: red; }
```

## I-H - Sélecteur de classe

Comme vu précédemment, on utilise le caractère "." comme sélecteur de classe.

Le sélecteur suivant représente un élément *h1* portant la class "referenceur".

```
h1.referenceur
```

## I-I - Sélecteur d'ID

Le sélecteur d'ID correspond au caractère "#". Dans l'exemple suivant, le sélecteur d'ID représente un élément *h1* dont l'attribut *id* a la valeur "referencement".

```
h1#referencement
```

## I-J - Pseudo-éléments et pseudo-classes

Les sélecteurs de pseudo-classes et de pseudo-éléments sont utilisés dans les scénarios où sélectionner la position de l'élément dans l'arbre HTML n'est pas suffisant. Pour utiliser les pseudo-classes et les pseudo-éléments on utilise le caractère ":".

### I-J-1 - La pseudo-classe :first-child

La pseudo-classe *:first-child* correspond au premier élément enfant d'un autre élément.

```
div > p:first-child { text-indent: 0; }
```

### I-J-2 - Les pseudo-classes d'ancre :link et :visited

La pseudo-classe *:link* s'applique aux liens qui n'ont pas été visités et la pseudo-classe *:visited* s'applique lorsque le lien a été visité par l'utilisateur.

### I-J-3 - Les pseudo-classes dynamiques :hover, :active et :focus

Les pseudo-classes dynamiques appliquent un changement de style en fonction d'une action utilisateur. La pseudo-classe *:hover* applique un style quand l'utilisateur désigne un élément (par exemple au survol de la souris), la pseudo-classe *:active* applique un style quand l'utilisateur active un élément (par exemple au moment où l'utilisateur presse le bouton de la souris et le relâche) et la pseudo-classe *:focus* applique un style quand un élément reçoit l'attention (par exemple quand l'utilisateur se positionne sur l'élément à l'aide du clavier).

```
a:link { color: red } /* lien non-visité */  
a:visited { color: blue } /* lien visité */  
a:hover { color: yellow } /* lien survolé */  
a:active { color: lime } /* lien activé */
```

### I-J-4 - La pseudo-classe de langue :lang

La pseudo-classe *:lang* permet de cibler un élément dont la langue correspond à une certaine valeur.

```
p:lang(en) { background-color: red; }
```

## I-J-5 - Le pseudo-élément :first-line

Le pseudo-élément `:first-line` associé à un paragraphe produit un style particulier sur la première ligne formatée d'un paragraphe.

```
p:first-line { text-transform: uppercase; }
```

## I-J-6 - Le pseudo-élément :first-letter

Le pseudo-élément `:first-letter` peut être employé pour faire des capitales initiales et des lettrines.

## I-J-7 - Les pseudo-éléments :before et :after

Les pseudo-élément `:before` et `:after` servent à insérer un contenu généré avant ou après celui d'un élément.

## Nouveautés CSS 3 sur les sélecteurs

### II-A - Sélecteur d'attribut

**3 nouveaux sélecteurs d'attribut sont introduits par la norme CSS 3 :**

- `[att^="val"]` : Représente un attribut dont la valeur commence exactement par le préfixe "val".
- `[att$=ident]` : Représente un attribut dont la valeur finit exactement par le suffixe "ident".
- `[att*="val"]` : Représente un attribut dont la valeur contient au moins une fois la sous-chaîne "val".

Le seul navigateur qui ne supporte pas les sélecteurs d'attribut CSS3 est IE6. IE7, IE8, Opera et les navigateurs basés les moteurs Webkit (Safari et Chrome) et Gecko (Firefox) supportent ces sélecteurs.

### II-B - Combinateur d'adjacence indirecte

Les combinateurs d'adjacence indirecte sont composés du caractère "~" séparant deux séquences de sélecteurs simples.

L'exemple suivant permet d'ajouter une bordure grise à toutes les images qui suivent une *div* particulière (l'image et la *div* doivent avoir le même parent).

```
div~img { border: 1px solid #ccc; }
```

Tous les navigateurs supportent le combinateur d'adjacence indirecte sauf votre navigateur favori : Internet Explorer 6.

### II-C - Pseudo-classes

Contient les plus grosses nouveautés de la norme CSS 3.

## II-C-1 - La pseudo-classe :nth-child()

Ce sélecteur vous permet de cibler les éléments en se basant sur leur position dans la liste des enfants de leur parent. Vous pouvez utiliser un numéro, une expression numérique ou les mots "odd" et "even" correspondant à impair et pair (parfait pour faire un style alternatif pour vos tableaux).

```
tr:nth-child(2n+1) /* correspond aux lignes paires */  
tr:nth-child(odd) /* pareil */
```

## II-C-2 - La pseudo-classe :nth-last-child()

Ce sélecteur ressemble beaucoup au sélecteur précédent sauf qu'il correspond au dernier enfant d'un élément parent.

```
tr:nth-last-child(-n+2) /* correspond aux 2 dernières lignes du tableau */
```

## II-C-3 - La pseudo-classe :last-child

Identique à *:nth-last-child(1)*. La pseudo-classe *:last-child* représente un élément qui est le dernier fils d'un autre élément.

## II-C-4 - La pseudo-classe :checked

Correspond aux éléments qui sont cochés.

## II-C-5 - La pseudo-classe :empty

Correspond aux éléments qui n'ont pas d'enfants ou qui sont vides.

## II-C-6 - La pseudo-classe de negation

La pseudo-classe de négation est une notation fonctionnelle prenant un sélecteur simple pour argument. Elle représente un élément qui n'est pas représenté par l'argument.

L'exemple ci-dessous permet d'afficher le texte des paragraphes qui n'ont pas la classe "lead" en noir.

```
p:not([class*="lead"]) { color: black; }
```

Les navigateurs basés sur le moteur Webkit ou Opera supportent toutes les nouveautés CSS 3 des sélecteurs de pseudo-classe. Firefox 2 et 3 supporte uniquement *:not(s)*, *:last-child*, *:only-child*, *:root*, *:empty*, *:target*, *:checked*, *:enabled* et *:disabled*, mais Firefox 3.5 aura un support complet des sélecteurs CSS 3. Internet Explorer n'a actuellement aucun support de ces pseudo-classes.

## II-D - Pseudo-élément

### II-D-1 - Les pseudo-éléments fragments d'éléments d'interface

Le nouveau pseudo-élément `::selection` s'applique à la portion du document qui a été mise en exergue par l'utilisateur.

L'exemple suivant permet de modifier la couleur de fond du texte en cours de sélection.

```
::selection { background-color: blue; }
```

### Pour en savoir plus

- **Module CSS3 : Sélecteurs W3C - Traduction officielle du brouillon de la norme**
- **Selectors Level 3 - W3C Working Draft**
- **Selectutorial - CSS selectors**
- **CSS3: Attribute selectors - CSS3.info**
- **CSS selectors and pseudo selectors browser compatibility**, sur cette page vous retrouverez en détails le support des sélecteurs par navigateur.