

# Les pseudo sélecteurs

---

Vous avez peut-être déjà vu certains mot-clé derrière des sélecteurs en CSS (hover, link, visited, active, first-child,...). Ce sont les pseudo sélecteurs, on en distingue de deux types: les **pseudo-elements** et les **pseudo-class**.

- **Pseudo-class**
  - Les liens
  - Focus
  - Pseudo-class et CSS Class
  - First-child
  - Last-child
  - First-of-type
  - Last-of-type
  - Nth-child
  - Nth-last-child
  - Toutes les pseudo-class
- **Pseudo-Element**
  - After & Before
  - First-letter & First-line
  - Selection
  - Tous les pseudo-element
- On combine tout!

## Pseudo-class

Les pseudo-class permet de définir l'état d'un élément. Par exemple:

- Le style d'un élément quand on le survol avec le curseur de la souris
- Le style d'un lien visité et non-visité
- Le style d'un élément quand il est "focus"

Il se note à la suite d'un sélecteur avec **:pseudo-class**

### Les liens

```
/* lien non-visité */
a:link {color: #FF0000;}

/* lien visité */
a:visited {color: #00FF00;}

/* survol avec le souris */
a:hover {color: #FF00FF;}

/* lien sélectionné */
a:active {color: #0000FF;}
```

Ces propriétés CSS vont permettre de modifier le comportement par défaut des styles des liens sur votre page.

- **link**: permet de gérer le style des liens tant qu'il n'ont pas été cliqué. Mettre une couleur différente de votre couleur de texte est recommandé pour attirer le visiteur à cliquer sur le lien.
- **visited**: permet de gérer le style des liens quand le visiteur à déjà cliqué dessus.
- **hover**: permet de gérer le style des liens lorsqu'on survole ce dernier avec le curseur de la souris.
- **active**: permet de gérer le style des liens quand ils sont cliqués.

## Focus

La pseudo-class `:focus` permet de définir le comportement d'un élément sélectionné par l'utilisateur généralement dans un formulaire.

```
input:focus {  
  background-color: yellow;  
}
```

## Pseudo-class et CSS Class

Il est tout à fait possible de combiner les classes CSS et les pseudo-classes.

```
a.important:hover{  
  color:red;  
}
```

## First-child

La pseudo-class `:first-child` permet de sélectionner spécifiquement le premier élément d'un groupe d'élément semblable.

```
p:first-child{  
  color:red;  
}
```

```
<p>Je serai en rouge</p>  
<p>Je n'aurai pas de couleur spécifique</p>  
<p>Je n'aurai pas de couleur spécifique</p>
```

Sélectionner la première balise `<em>` dans toutes les `<p>`

```
<style>
  p em:first-child{
    color:red;
  }
</style>
<p>Je suis une phrase, <em>moi je serai en rouge</em> et <em>moi, comme je
suis le deuxième élément en italique, je ne le serai pas</em></p>
```

### Sélectionner toutes les balises `<em>` dans le "first-child" `<p>`

```
<style>
  p:first-child i{
    color: red;
  }
</style>
<p>Salut <em>je serai en rouge</em> et <em>moi aussi</em></p>
<p>Salut <em>je serai PAS en rouge</em> et <em>moi NON PLUS</em></p>
```

### Last-child

Fonctionne comme pour `:first-child` mais au lieu de sélectionner le premier enfant de l'élément, il sélectionnera le dernier.

### First-of-type

La pseudo-class `:first-of-type` va sélectionner tous les éléments dont c'est leur première apparition.

```
<style>
  article :first-of-type{
    background-color: red;
  }
</style>
<article>
  <div>Je suis la première div</div>
  <div>Je suis le premier <span>span</span></div>
  <p>Je suis le premier paragraphe</p>
  <div>Et moi je suis la dernière div</div>
</article>
```

### Last-of-type

Fonctionne comme `:first-of-type` mais sélectionnera le dernier élément qui apparaît.

### Nth-child

Permet de sélectionner un élément par rapport à sa position dans un groupe d'élément semblable.

```
a:nth-child(2){  
  color:orange;  
}
```

Dans cet exemple on va sélectionner le deuxième lien (a) disponible

Ce sélecteur prend un paramètre supplémentaire entre parenthèse. Il peut s'agir des mots-clés: **odd** (impair) ou **even** (pair) ou d'un chiffre. Il est également possible d'y introduire des formules.

```
a:nth-child(2n+1){  
  color:orange  
}
```

Voilà comment est interprété la formule:

$(2 \times 0) + 1 = 1 = 1\text{st Element}$

$(2 \times 1) + 1 = 3 = 3\text{rd Element}$

$(2 \times 2) + 1 = 5 = 5\text{th Element}$

Cela revient pareil à indiquer **odd**, mais il est possible d'utiliser des formules différentes pour d'autres sélections, par exemple:

```
a:nth-child(3n+3){  
  color: green;  
}
```

$(3 \times 0) + 3 = 3\text{rd Element}$

$(3 \times 1) + 3 = 6\text{th Element}$

$(3 \times 2) + 3 = 9\text{th Element}$



Il est possible de raccourcir cette formule par **3n** pour sélectionner, du coup, tous les 3 éléments.



Le + dans la formule permet de démarrer le compte en commençant par le chiffre qui suit. Par exemple: **3n+5** va sélectionner tous les 3 éléments en commençant par le cinquième.

[En apprendre plus sur les formules](#)

C'est un peu compliqué ces formules, c'est bien de savoir que ça existe pour quand on les rencontres dans du code ou pour réaliser des styles bien spécifique.

### Nth-last-child

Fonctionne comme **:nth-child** mais compte en commençant par la fin.

## Toutes les pseudo-class

Il y a évidemment énormément d'autres pseudo-class qui ont chacune leurs utilités. Nous venons d'en voir quelques unes qui seront celle que vous utiliserez le plus fréquemment dans vos travaux.

Voici [un lien vers la MDN et toutes les pseudo-class](#)

## Pseudo-Element

Les pseudo-element sont des mots-clés que l'on rajoute à la fin d'un sélecteur pour sélectionner une partie précise de l'élément.

Il se note à la suite d'un sélecteur avec `::pseudo-element`

### After & Before

Permet de sélectionner le début (before) ou la fin (after) d'un élément. Cela sert notamment à ajouter un style à un élément grâce à la propriété `content`.

```
a::before{
  content:"This is a link -> "
}
a::after{
  content:" <- Click on me";
}
```



Le contenu ajouté sera en `inline` dans le bloc

### First-letter & First-line

Permet de sélectionner la première lettre ou ligne d'un élément.

```
p::first-letter {
  font-size: 230%;
}
p::first-line {
  color: red;
}
```

### Selection

Permet de styliser la sélection d'un élément par l'utilisateur.

```
::selection {
  color: red;
  background: yellow;
}
```



Tous les éléments sélectionnés ici seront en rouge et sur fond jaune.

## Tous les pseudo-element

Il existe encore quelques autres pseudo-element, mais rien de bien utile dans le cadre de ce cours, voici tout de même [un lien vers la doc](#).

## On combine tout!

Il est tout à fait possible de combiner plusieurs pseudo-sélecteur.

```
p:nth-child(even):hover::first-letter {  
  color:red;  
}
```



Ici on sélectionne la première lettre de tous les paragraphe pair et on applique une couleur dessus quand on passe notre souris par dessus cet élément (tout le paragraphe). C'est un peu tiré par les cheveux, mais ça fonctionne.



[Retour au sommaire du cours](#)