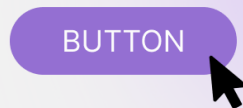


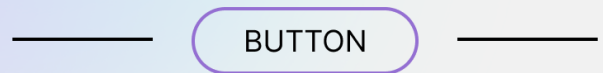
:

pseudo-class



::

pseudo-element



CSS

Meet the

PSEUDOs

Sommaire

Meet the PSEUDO CLASS & ELEMENTS	0
Sommaire	1
INTRODUCTION.....	2
1. Sélecteurs de pseudo-class liés aux liens	3
La pseudo-classe : <code>link</code> va elle nous permettre au contraire de cibler tous les liens non visités et de leur appliquer des styles particuliers en CSS	3
<code>:visited</code>	3
2. Pseudo-sélecteurs de classe liés aux Input et aux link	5
3. Pseudo-class basés sur la position/le nombre	7
4. Pseudo-class relationnel	9
5. Pseudo-sélecteurs de classe / éléments liés au texte	10
6. Pseudo-éléments liés au contenu	11

INTRODUCTION

En CSS, les pseudo-classes et les pseudo-éléments sont deux types de mots-clés que vous pouvez combiner avec des sélecteurs. Ils sont utilisés pour cibler l'état de l'élément ou des parties spécifiques d'un élément.

Les sélecteurs de la classe Pseudo sont des sélecteurs CSS avec un « **:** » qui les précède.

Vous en connaissez probablement très bien quelques-uns. Comme le [hover](#):

```
a:hover {  
  /* Yep, hover is a pseudo class */  
}
```

Syntaxe

- Le deux points simple **:** fait référence aux pseudo-classes
- Le double deux points **::** fait référence aux pseudo-éléments

Ils sont extrêmement utiles dans diverses situations.

Certains d'entre eux sont CSS3, d'autres CSS2...

Il existe de nombreuses pseudo-classes en CSS.

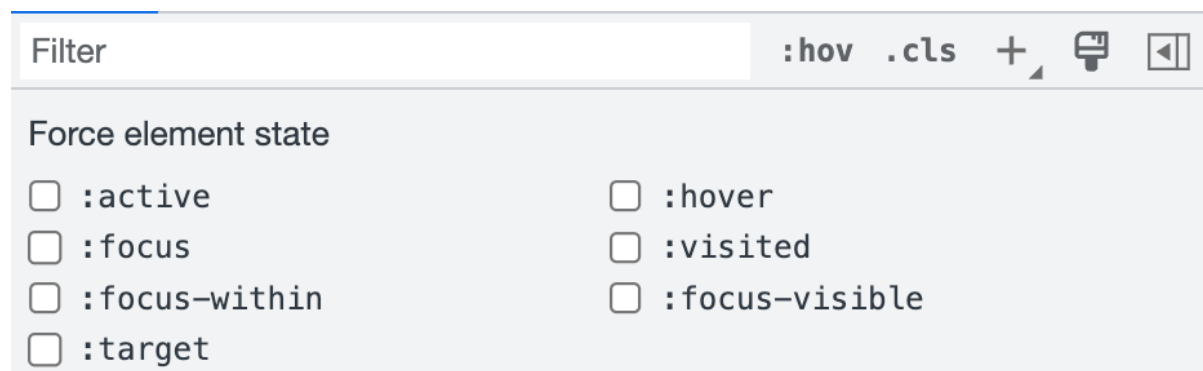
Dans cette leçon, nous n'allons étudier en détail que les plus utilisées car toutes les pseudo-classes s'utilisent de manière similaire (la syntaxe sera toujours la même).

Ainsi, il vous suffira de comprendre comment appliquer une pseudo-classe pour savoir toutes les utiliser, à condition bien évidemment de savoir à quoi correspond chaque pseudo-classe.

Pour cela, je vais également vous fournir un tableau récapitulatif de toutes les pseudo-classes actuellement disponibles afin que vous puissiez piocher dedans selon vos besoins

Jetons un bref coup d'œil à chacun d'eux.

1. SÉLECTEURS DE PSEUDO-CLASS LIÉS AUX LIENS



Vue de l'inspecteur chrome

[:link](#)

La pseudo-classe `:link` va elle nous permettre au contraire de cibler tous les liens non visités et de leur appliquer des styles particuliers en CSS

[:visited](#)

Sélectionne les liens qui ont déjà été visités par le navigateur actuel.

[:hover](#)

La pseudo-classe `:hover` va nous permettre d'appliquer des styles à un élément HTML uniquement lorsque celui-ci est survolé par la souris d'un utilisateur.

Cette pseudo-classe peut être utilisée avec la majorité des éléments HTML mais on l'utilisera généralement pour appliquer des styles différents à des liens lorsqu'un utilisateur passe sa souris dessus.

[:active](#)

La pseudo-classe CSS `:active` va elle nous permettre de modifier les styles d'un élément HTML lors d'un état très particulier qui est le moment où l'on clique sur

l'élément. Pour bien visualiser cet état, je vous conseille de rester cliqué sur l'élément en question le temps de voir les changements de style. Une nouvelle fois, en pratique, cette pseudo-classe va généralement être utilisée pour modifier l'apparence des liens au moment du clic.

Par exemple, pour l'état « pressé » d'un lien de type bouton ou pour [que tous les liens ressemblent plus à un bouton](#).

Exemple de pseudo-class:

👉 <https://codepen.io/pierregiraud/pen/bOpxNg>

2. PSEUDO-SÉLECTEURS DE CLASSE LIÉS AUX INPUT ET AUX LINK

:focus

Définir les styles de survol (`:hover`) pour les liens est une bonne chose, mais cela n'aide pas ceux qui ont utilisé la navigation au clavier pour atteindre le lien. `:focus` sélectionne les liens qui sont au centre de l'attention du clavier.

Ceci n'est pas limité aux liens, mais peut être utilisé (et devrait l'être) pour les input et les zones de texte également.

Certains vous diront de définir un style `:focus` pour tout ce qui a un style `:hover`

:target

La pseudo classe `target` est utilisée en conjonction avec les ID, et correspond lorsque la balise de hachage de l'URL actuelle correspond à cet ID.

Ainsi, si vous êtes sur l'URL `www.yoursite.com/#home`, le sélecteur `#home:target` correspondra. Cela peut être extrêmement puissant.

Par exemple, vous pouvez créer une zone à onglets où les onglets sont liés à des balises de hachage et où les panneaux sont "activés" en faisant correspondre les sélecteurs `:target`

:enabled

Sélectionne les inputs qui sont activées par défaut et prêtes à être utilisées.

:disabled

Sélectionne les inputs qui ont l'attribut `disabled`.

Beaucoup de navigateurs rendent l'input grise, vous pouvez contrôler cela avec ce sélecteur.

:checked

Sélectionne les cases à cocher qui sont, attendez, cochées.

:indeterminate

Sélectionne les **boutons radio** qui ne sont ni choisis ni non choisis (comme lorsqu'une page se charge avec des choix de boutons radio mais qu'aucune valeur par défaut n'est définie).

:required

Sélectionne les inputs avec l'attribut requis.

:optional

Sélectionne les entrées qui n'ont pas l'attribut requis.

:read-only / :read-write

Sélectionne les éléments en fonction d'une combinaison d'attributs **readonly** et **disabled**.

3. PSEUDO-CLASS BASÉS SUR LA POSITION/LE NOMBRE

`:root`

Sélectionne l'élément qui se trouve à la racine du document.

`:first-child`

Sélectionne le premier élément à l'intérieur d'un parent.

`:last-child`

Sélectionne le dernier élément d'un parent.

`:nth-child()`

Pour pouvoir cibler n'importe quel élément HTML en fonction de sa place dans le document, nous allons également pouvoir utiliser les sélecteurs CSS `:nth-child(n)` et `:nth-of-type(n)` qui vont nous permettre de sélectionner un élément HTML qui est le *nième* enfant de son parent ou un élément qui est le *nième* enfant de ce type de son parent.

👉 CodePen : <https://codepen.io/zakkain/pen/AzKxdx?editors=1100>

`:first-of-type`

Sélectionne le premier élément de ce type à l'intérieur d'un parent. Ainsi, si vous avez deux `div` contenant chacune un paragraphe, une image, un paragraphe et une image, la div `img:first-of-type` sélectionnera le premier élément du premier div et le premier élément du second div. Le div `img:first-of-type` sélectionnerait la première image à l'intérieur de la première div et la première image à l'intérieur de la deuxième div.

`:last-of-type`

Identique à la précédente, mais sélectionnerait la dernière image à l'intérieur de la première div et la dernière image à l'intérieur de la deuxième div.

`:nth-last-of-type()`

Fonctionne comme `:nth-of-type`, mais compte à partir du bas au lieu du haut.

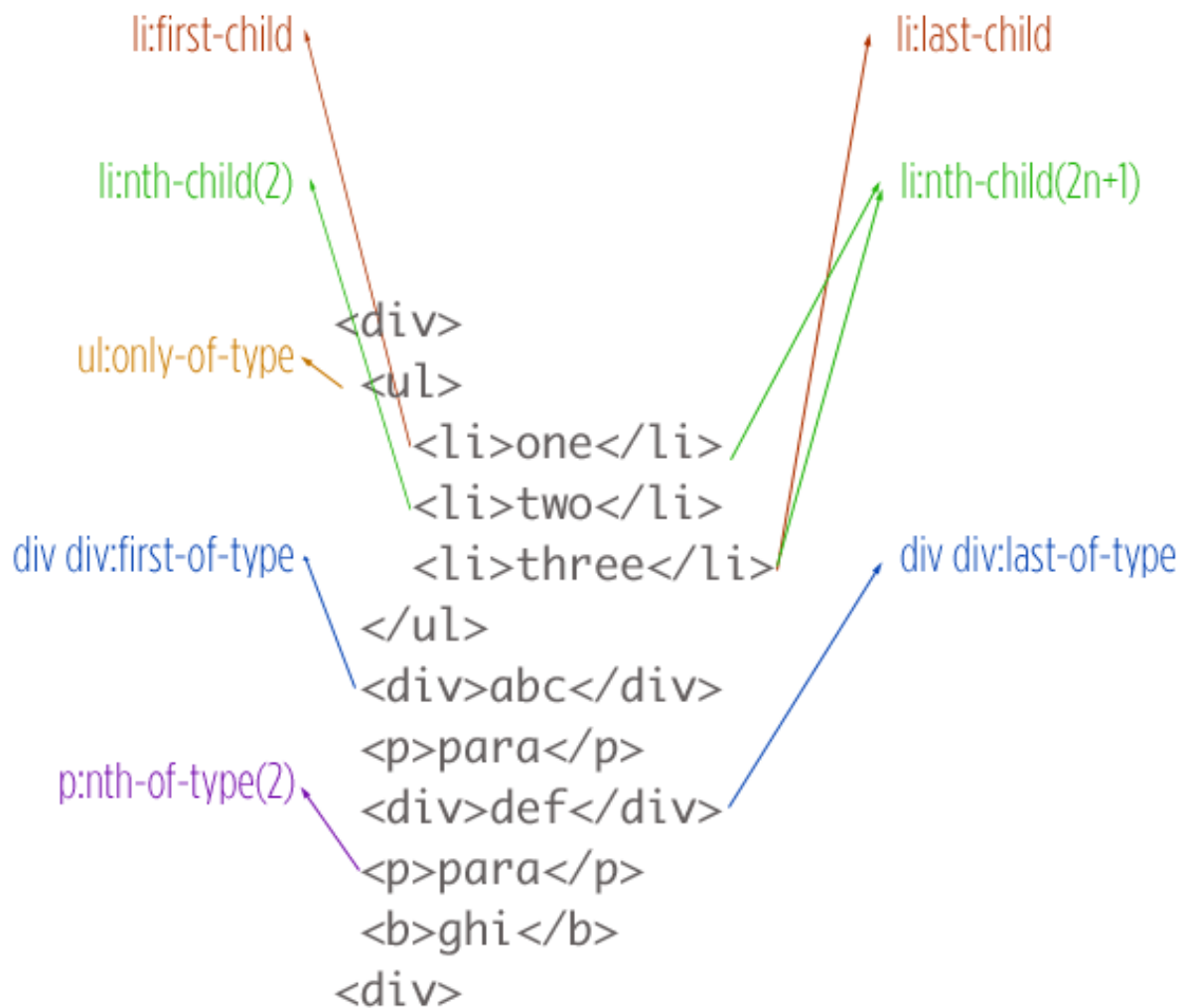
`:nth-last-child()`

Fonctionne comme `:nth-child`, mais compte à partir du bas au lieu du haut.

`:only-of-type`

Ne sélectionne que si l'élément est le seul de son genre dans le parent actuel.

Illustration des pseudo classes



👉 CodePen <https://codepen.io/pierregiraud/pen/YdqOOW>

4. PSEUDO-CLASS RELATIONNEL

:not()

Supprime les éléments d'un ensemble existant qui correspondent au sélecteur contenu dans le paramètre de `:not()`.

Ainsi, par exemple, toutes les divs sauf celles ayant une classe "music" = `div:not(.music)`.

La spécification indique que les sélecteurs `:not` ne peuvent pas être imbriqués, mais qu'ils peuvent être enchaînés.

Certains navigateurs (Firefox) prennent également en charge les sélecteurs séparés par des virgules en tant que paramètre du sélecteur, bien qu'il soit plus sûr de les enchaîner. Également utile avec les sélecteurs d'attributs, par exemple `input:not([disabled])`.

:empty

Sélectionne les éléments qui ne contiennent ni texte ni éléments enfants.

:is

La `:is` pseudo-classe prend une liste d'arguments de sélecteur et fait correspondre tous les éléments qui s'appliquent à n'importe quel sélecteur de la liste.

Voici la syntaxe :

```
button:is(:hover, :focus, .active) {  
  background-color: black;  
  color: white;  
}
```

⚠ `:is` ne prend pas en charge les pseudo-éléments (par ex: `::after`)

Vous pouvez effectuer de nombreuses sélections avancées avec la `:is` pseudo-classe. Cette classe vous permet d'écrire de longs sélecteurs de manière plus courte et plus facile à lire.

5. PSEUDO-SÉLECTEURS DE CLASSE / ÉLÉMENTS LIÉS AU TEXTE

`::first-letter` - Sélectionne la première lettre du texte dans l'élément.
Utilisation typique : majuscules.

`::first-line` - Sélectionne la première ligne du texte dans l'élément. Utilisation typique : mettre la première phrase en petites capitales pour attirer l'attention sur la typographie.

`:lang` - Ce pseudo-sélecteur figure dans la spécification CSS3.
Il s'applique à tout ce qui a un élément ou est un descendant d'un élément avec un attribut `lang` correspondant. Par exemple, `:lang(fr)` correspondra à n'importe quel paragraphe, même sans attribut `lang`, si le corps de l'élément parent a pour attribut `lang="fr"`.



Note :

Vous pouvez enchaîner des pseudo-sélecteurs tout comme vous pouvez enchaîner des sélecteurs de classe et d'ID. C'est particulièrement utile ici, lorsque nous examinons `:first-letter` et `:first-line`.

Vous ne voudrez probablement pas mettre en surbrillance tous les paragraphes de la page, mais seulement le premier, donc : `p:first-child:first-letter { }`

6. PSEUDO-ÉLÉMENTS LIÉS AU CONTENU

::before

Permet d'ajouter du contenu **avant** un certain élément.

Par exemple, l'ajout d'une citation d'ouverture avant une citation en bloc ou d'une image précédente pour distinguer un paragraphe particulier.

::after

Permet d'ajouter du contenu **après** un certain élément.

Par exemple, un guillemet fermant à une citation en bloc. Également utilisé couramment pour le clearfix, où un espace vide est ajouté après l'élément qui efface le flotteur sans qu'il soit nécessaire de recourir à un balisage HTML supplémentaire.

Syntaxe

```
/* PSEUDO-ELEMENTS */
h1::before,
h1::after {
  content: "";
  height: 3px;
  background: black;
  flex: 1;
  margin: 0.2em;
}
```

Source : <https://codepen.io/nataliepina/pen/qBJqpgq>

Liste complète des pseudo-classes et définitions

Vous pourrez trouver ci-dessous toutes les pseudo-classes avec la description de leur comportement et la version CSS dans laquelle elles sont passées comme recommandation pour référence.

Les pseudo-classes dont la version est « 3/4 » sont celles dont la définition initiale a été posée avec le CSS3 mais dont la définition va certainement être modifiée avec la prochaine version du CSS.

De plus, notez que certaines pseudo-classes ont été créées pour ne fonctionner qu'avec certains éléments en particulier et notamment avec des éléments de formulaire que nous étudierons plus tard.

Sélecteur	Description	Version CSS
E:link	Sélectionne tout élément E représentant l'ancree d'un lien non visité jusqu'à présent	CSS1
E:visited	Sélectionne tout élément E représentant l'ancree d'un lien déjà visité	CSS1
E:active	Sélectionne un élément E au moment où il est cliqué	CSS1
E:hover	Sélectionne un élément E lorsque le curseur de la souris passe dessus	CSS2
E:focus	Sélectionne un élément E qui a le focus (dans lequel le curseur de la souris est placé)	CSS2
E:first-child	Sélectionne tout élément E étant le premier enfant de son parent	CSS2

E:lang(fr)	Sélectionne tout élément E dont l'attribut langage possède la valeur « fr »	CSS2
E:target	Sélectionne un élément E contenant une ancre qui vient d'être cliquée à partir d'un lien ancre	CSS3
E:enabled	Sélectionne tout élément E avec lequel l'utilisateur peut interagir et qui est activé	CSS3
E:disabled	Sélectionne tout élément E avec lequel l'utilisateur peut interagir et qui est désactivé	CSS3
E:root	Sélectionne un élément E racine du document	CSS3
E:empty	Sélectionne tout élément E qui ne possède pas d'enfant (ni de noeud de type texte)	CSS3
E:nth-child(n)	Sélectionne tout élément E étant le n-ième enfant de son parent	CSS3
E:nth-last-child(n)	Sélectionne tout élément E étant le n-ième enfant de son parent en comptant les enfants à partir du dernier	CSS3
E:checked	Sélectionne tout élément E de type input coché au sens large (checked ou selected)	CSS3
E:last-child	Sélectionne tout élément E étant le dernier enfant de son parent	CSS3
E:only-child	Sélectionne tout élément E qui est le seul enfant de son parent	CSS3
E:nth-of-type(n)	Sélectionne tout élément E étant le n-ième enfant d'un certain type par rapport à son parent	CSS3
E:nth-last-of-type(n)	Sélectionne tout élément E étant le n-ième enfant d'un certain type par rapport à son parent en comptant à partir de la fin	CSS3

E:first-of-type	Sélectionne tout élément E premier enfant de son type par rapport à son parent	CSS3
E:last-of-type	Sélectionne tout élément E dernier enfant de son type par rapport à son parent	CSS3
E:only-of-type	Sélectionne tout élément E seul enfant de son type par rapport à son parent	CSS3
E:read-write	Sélectionne tout élément E de type input avec lequel l'utilisateur peut interagir (comme un champ dans lequel il peut écrire par exemple)	3-UI/4
E:read-only	Sélectionne tout élément E de type input avec lequel l'utilisateur ne peut pas interagir (éléments possédant un attribut disabled par exemple)	3-UI/4
E:placeholder-shown	Sélectionne tout élément E qui affiche actuellement la valeur de son attribut placeholder	3-UI/4
E:default	Sélectionne un élément E dans une liste ou un groupe qui est l'élément défini par défaut	3-UI/4
E:valid	Sélectionne tout élément E de type input dont la valeur est évaluée comme valide (dont la valeur possède une forme correspondant à ce qui est attendu)	3-UI/4
E:invalid	Sélectionne tout élément E de type input dont la valeur est évaluée comme invalide (valeur ne correspondant pas à ce qui est attendu)	3-UI/4
E:in-range	Sélectionne tout élément E de type input dont la valeur fournie se situe dans une fourchette de valeurs prédéfinies	3-UI/4
E:out-of-range	Sélectionne tout élément E de type input dont la valeur fournie se situe en dehors d'une certaine fourchette de valeurs prédéfinies	3-UI/4
E:required	Sélectionne tout élément E de type input dont la valeur doit être renseignée (élément possédant un attribut required)	3-UI/4

E:optional	Sélectionne tout élément E de type input dont la valeur ne doit pas obligatoirement être renseignée	3-UI/4
E:not(E1, .c1)	Sélectionne tout élément E qui n'est pas de type E1 et qui ne possède pas d'attribut class= »c1"	3/4

Testeur de nth-child

📄 <https://css-tricks.com/examples/nth-child-tester/>

SOURCES :

- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes>
- <https://css-tricks.com/almanac/pseudo-selectors/n/nth-child/>
- <https://dillionmegida.com/p/pseudo-elements-vs-pseudo-classes-in-css/>
- <https://www.freecodecamp.org/news/css-is-pseudo-class-explained/>
- <https://css-tricks.com/almanac/pseudo-selectors/>
- https://developer.mozilla.org/fr/docs/Web/CSS/Selector_list

Happy styling!