

CSS SELECTORS



Les Sélecteurs CSS pour le Web Scraping

Utilité des Sélecteurs CSS

Les sélecteurs CSS permettent de **cibler précisément des éléments HTML**. En web scraping, ils permettent de **naviguer dans le DOM** comme un navigateur le ferait.

```
<div class="price">$19.99</div>
```

```
soup.select("div.price")  
# → [<div class="price">$19.99</div>]
```

Sélecteurs de base

Sélecteur	Signification	Exemple
tag	Sélectionne tous les éléments de ce type	p , div , a
.class	Sélectionne les éléments avec une classe	.title
#id	Sélectionne un élément avec un ID unique	#main-header

```
<p class="title" id="titre1">Bienvenue</p>
```

✓ .title

✓ #titre1

✗ title → Faux, ce n'est pas un tag HTML.

Sélecteurs d'attributs

Sélecteur	Signification	Exemple
-----------	---------------	---------

[attr]	L'élément possède cet attribut	[href]
[attr="valeur"]	Attribut égal à une valeur précise	[type="submit"]
[attr^="valeur"]	L'attribut commence par	[src^="https"]
[attr\$="valeur"]	L'attribut finit par	[src\$=".jpg"]
[attr*="valeur"]	L'attribut contient la valeur	[href*="login"]

```
<a href="/login" class="btn">Connexion</a>
```

✓ [href="/login"]

✓ [href*="login"]

✓ a.btn[href^="/"]

✗ .btn[href=login] → Mauvais format, pas de guillemets

Combinateurs CSS

Les combinateurs définissent **des relations entre les éléments**.

◆ A B — Descendant

Sélectionne tous les B **dans** A (même profondément imbriqué).

```
<div class="product">
  <span>Nom</span>
  <div><span>Prix</span></div>
</div>
```

.product span → cible les 2

.product > span → ne sélectionne que le premier , pas le deuxième (car il est imbriqué dans un <div>)

◆ A > B — Enfant direct

```
<ul>
  <li>Premier</li>
```

```
<li>Deuxième</li>  
</ul>
```

ul > li → sélectionne les qui sont enfants directs de

```
<ul>  
  <div>  
    <span>  
      <li>Pas direct</li>  
    </div>  
</ul>
```

ul > li → ne sélectionne rien ici (car est dans un <div>)

◆ A + B — Frère direct immédiat

```
<h2>Titre</h2>  
<p>Intro</p>
```

h2 + p → sélectionne le <p> juste après <h2>

```
<h2>Titre</h2>  
<div><p>Intro</p></div>
```

h2 + p → ne sélectionne rien (le <p> n'est pas directement après <h2>)

◆ A ~ B — Tous les frères suivants

```
<h2>Titre</h2>  
<p>Intro</p>  
<p>Résumé</p>
```

h2 ~ p → sélectionne tous les <p> qui suivent <h2> (au même niveau)

```
<section>
  <h2>Titre</h2>
</section>
<p>Hors section</p>
```

h2 ~ p → ne sélectionne pas le <p> car il n'est ****pas frère**** (différent parent)

Pseudo-classes

Sélecteur	Signification	Exemple HTML
<code>:first-child</code>	Premier enfant de son parent	<code>li:first-child</code>
<code>:last-child</code>	Dernier enfant	<code>li:last-child</code>
<code>:nth-child(n)</code>	Nième enfant	<code>tr:nth-child(2)</code>
<code>:nth-of-type(n)</code>	Nième enfant du même type	<code>p:nth-of-type(2)</code>
<code>:not(selector)</code>	Exclut des éléments	<code>div:not(.hidden)</code>

```
<ul>
  <li>1</li>
  <li>2</li>
  <li>3</li>
</ul>
```

`li:first-child` → 1
`li:nth-child(2)` → 2

```
<ul>
  <div>Texte</div>
  <li>Premier li</li>
</ul>
```

`li:first-child` → ❌ Faux (le li n'est pas le ****premier enfant****)

Utilisez plutôt :

`li:nth-of-type(1)`

Cas concrets

Liste de produits

```
<div class="product">
  <h2 class="title">Ordinateur</h2>
  <span class="price">$999</span>
</div>
```

```
soup.select("div.product h2.title")
soup.select(".product > .price") # seulement si .price est enfant direct
```

```
<div class="product">
  <div class="details">
    <span class="price">$999</span>
  </div>
</div>
```

`.product > .price` → ❌ ne fonctionne pas
`.product .price` → ✅ fonctionne



Fiche Mémo

♦ Basiques

tag	— div, a, span
.class	— .price
#id	— #header

- ♦ Attributs

[attr] — [href]
[attr="v"] — [type="submit"]
[attr^="v"] — [src^="https"]
[attr\$="v"] — [src\$=".jpg"]
[attr*="v"] — [href*="product"]

- ♦ Combinateurs

A B — Tous les B dans A
A > B — B enfants directs de A
A + B — B juste après A (frère direct)
A ~ B — B frères de A, plus loin

- ♦ Pseudo-classes

:first-child — Premier enfant
:last-child — Dernier enfant
:nth-child(n) — Enfant n
:nth-of-type(n) — n-ième enfant du même type
:not(sel) — Exclusion

- ♦ Sélecteurs complexes

div.class[attr*="value"]
a[href^="https"]:not(.external)

Docs :

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_selectors

Exos :

<https://css-selector.netlify.app/>