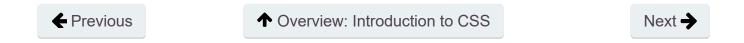


# Combinators and selector lists



In our final article on selectors, we'll explore combinators and groups of selectors — two ways of combining multiple selectors together for further useful selection capabilities.

#### Combinators and selector lists &

Using one selector at a time is useful, but can be inefficient in some situations. CSS selectors become even more useful when you start combining them to perform fine-grained selections.

CSS has several ways to select elements based on how they are related to one another. Those relationships are expressed with *combinators* as follows (A and B represent any selector seen above):

Name	Syntax	Selects
Selector list	А, В	Any element matching A and/or B (see Groups of selectors on one rule, below - <b>Group of Selectors</b> is not considered to be a combinator).
Descendant combinator	A B	Any element matching B that is a <i>descendant</i> of an element matching A (that is, a child, or a child of a child, etc.). the combinator is one or more spaces or dual greater than signs.
Child combinator	A > B	Any element matching B that is a direct child of an element matching A.
Adjacent sibling combinator	A + B	Any element matching B that is the next <i>sibling</i> of an element matching A (that is, the next child of the same parent).
General sibling combinator	A ~ B	Any element matching B that is one of the next <i>siblings</i> of an element matching A (that is, one of the next children of the same parent).

### Combinators example &



Let's look at an example with all of this working together:

```
1
 <thead>
 3
  Product
4
  Qty
5
  Price
```

```
6
7
   </thead>
   <tfoot>
8
    9
     Total:
10
11
     148.55
12
    </tfoot>
13
   14
15
    Lawnchair
16
     1
17
     137.00
18
19
    20
    Marshmallow rice bar
21
     2
22
     1.10
23
    24
25
    Book
26
     1
27
     10.45
28
29
    30
  31
```

Then, let's use the following style sheet:

```
table {
2
      font: 1em sans-serif;
3
      border-collapse: collapse;
4
      border-spacing: 0;
5
6
7
    /* All s within a  and all s within a 
8
       Comma is not a combinator, it just allows you to target
9
       several selectors with the same CSS ruleset */
10
    table td, table th {
11
      border: 1px solid black;
12
      padding: 0.5em 0.5em 0.4em;
13
14
15
    /* All s within <thead>s that are within s */
16
    table thead th {
17
      color: white;
18
      background: black;
19
20
21
22
    /* All s preceded by another ,
       within a , within a  */
23
    table tbody td + td {
24
      text-align: center;
25
26
27
    /* All s that are a last child,
28
      within a , within a  */
29
    table tbody td:last-child {
30
31
      text-align: right;
```

```
32
33
    /* All s, within a <tfoot>s, within a  */
34
    table tfoot th {
35
      text-align: right;
36
      border-top-width: 5px;
37
      border-left: none;
38
      border-bottom: none;
39
40
41
    /* All s preceded by a , within a  */
42
    table th + td {
43
      text-align: right;
44
      border-top-width: 5px;
45
      color: white;
46
      background: black;
47
48
49
    /* All pseudo-elements "before" s that are a last child,
50
       appearing within elements with a class of "with-currency" that
51
       also have an attribute "lang" with the value "en-US" */
52
    .with-currency[lang="en-US"] td:last-child::before {
53
      content: '$';
54
55
56
    /* All pseudo-elements "after" s that are a last child,
57
       appearing within elements with the class "with-currency" that
58
       also have an attribute "lang" with the value "fr" */
59
    .with-currency[lang="fr"] td:last-child::after {
60
      content: ' €';
61
```

Product	Qty	Price
Lawnchair	1	\$137.00
Marshmallow rice bar	2	\$1.10
Book	1	\$10.45
1	\$148.55	

#### Active learning: Writing your own combinators



The above example was designed to show the sort of complexity you can start to achieve with combinators. In this active learning, we will get you to write some of your own, more simple selectors that include combinators. In this exercise, you need to add a selector to rules 2-4, to:

- 1. Style links, but only links that are inside the unordered list.
- 2. Style links inside the unordered list, but only when they are being hovered over.
- 3. Style only the paragraph that comes directly after the top-level heading.

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the Show solution button to see a potential answer.

#### III WIL IIIPUL

```
    <a href="#">Home</a>
    <a href="#">Portfolio</a>
    <a href="#">About</a>

<h1>Welcome to my website</h1>
Hello, and welcome! I hope you enjoy your time here.
```

# **CSS Input**

```
ul {
  padding: 0;
  list-style-type: none;
}

{
  text-decoration: none;
  display: block;
  color: black;
  hackground-color: pod:
```

# **Output**

Home Portfolio About

# Welcome to my website

Hello, and welcome! I hope you enjoy your time here.

Reset

Show solution

#### Groups of selectors on one rule $\mathscr{O}$

You have seen multiple examples of this in action already, but let's spell it out for clarification. You can write groups of selectors separated by commas to apply the same rule to multiple sets of selected elements at once. For example:

```
1  p, li {
2  font-size: 1.6em;
3  }
```

Or this:

```
h1, h2, h3, h4, h5, h6 {
font-family: helvetica, 'sans serif';
}
```

**①** Important: If a single selector is not supported by the browser, then the whole selector list gets ignored.

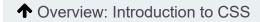
The :is() pseudo-class is not subject to this limitation, but isn't yet widely supported at the time of writing.

#### What's next $\mathcal{S}$

Congratulations — you've come to the end of our rather long journey into learning about CSS selectors. Even the most skilled web developers are still amazed by what's possible using selectors. However, don't feel bad if you can't remember all the options — bookmark the main selectors page and refer back to it when you need to.

In our next article, we'll turn to another really important fundamental CSS topic — the kind of values properties can have, and what units are involved in expressing the length, color, or any other values you want. Let's explore CSS values and units.







#### In this module §

- How CSS works
- CSS syntax
- Selectors
- Simple selectors
- Attribute selectors

- Pseudo-classes and pseudo-elements
- Combinators and multiple selectors
- CSS values and units
- Cascade and inheritance
- The box model
- Debugging CSS
- Fundamental CSS comprehension