

[Back to Class Four page »](#)

# CSS Selectors

The following is a list of the most common and well-supported CSS selectors. There are many, many more, but these are the ones you should know well.

- [Element Type Selectors](#)
- [Descendant Selectors](#)
- [Class selectors](#)
- [Id Selectors](#)
- [Child Selectors](#)
- [Adjacent sibling selectors](#)
- [Pseudo Selectors](#)
- [Universal Selectors](#)
- [Additional Reading](#)

## Element Type Selectors

The most basic CSS selectors are Element Type Selectors. That's a fancy name for simply **using an HTML tag**, without the angle braces.

We've used this selector extensively already.

For example, if we wanted to make **all paragraphs** have green text, we would use the following CSS rule:

```
p { color: green; }
```

- [See Demo](#)
- 

## Descendant Selectors

Match an element that is a descendant of another element.

This uses two separate selectors, separated by a space.

For example, if we wanted **all emphasized text in our paragraphs** to be green text, we would use the following CSS rule:

```
p em { color: green; }
```

- [See Demo](#)
- 

## Class Selectors

Match an element that has the specified class.

To match a specific `class` attribute, we always start the selector with a period, to signify that we are looking for a `class` value. The period is followed by the class attribute value we want to match.

For example, if we wanted **all elements with a class of "highlight"** to have a different background color, we would use the following CSS rule:

```
.highlight { background-color: #ffcccc; }
```

- [See Demo](#)
- 

## Id Selectors

Match an element that has the specified id.

To match a specific id attribute, we always start the selector with a hash symbol (#), to signify that we are looking for an id value. The hash is followed by the id attribute value we want to match. Remember, we can only use the same id attribute value once, so the id selector will always only match one element in our document.

For example, if we wanted **the element with an id of "content"**, we would use the following CSS rule:

```
#content { border: 2px solid green; }
```

- [See Demo](#)
- 

## Child selectors

Match an element that is an immediate child of another element.

For example, if we wanted **all emphasized text in our paragraphs's** to have green text, **but not emphasized text in other elements**, we would use the following CSS rule:

```
p > em { color: green; }
```

- [See Demo](#)

Note: This selector does not work in Internet Explorer 6

---

## Adjacent sibling selectors

Match an element that is immediately after another element, but not a child of it.

For example, if we wanted **all paragraphs that immediately followed an h4** to have green text, **but not other paragraphs**, we would use the following CSS rule:

```
h4 + p { color: green; }
```

- [See Demo](#)

Note: This selector does not work in Internet Explorer 6

---

## Pseudo Selectors

### An Aside About Link States

Anchor elements are special. You can style the <a> element with an Element Type Selector, but it might not do exactly what you expect. This is because links have different states, that relate to how they are interacted with. The four primary states of a link are: link, visited, hover, active.

Pseudo selectors come in different sizes and shapes. By far the most common pseudo selectors are used to style our links. There are four different pseudo selectors to be used in conjunction with links:

:link

A link that has not been previously visited (visited is defined by the browser history)

:visited

A link that has been visited

:hover

A link that the mouse cursor is "hovering" over

:active

A link that is currently being clicked

```
a:link          { color: red }           /* unvisited links      */
a:visited       { color: blue }          /* visited links       */
a:hover         { color: green }         /* user hovers         */
a:active        { color: lime }          /* active links        */
```

For reasons of browser compatibility, you should always specify the pseudo selectors in this order. An easy way to remember this is by using the mnemonic: "LoVe HA!".

- [See Demo](#)

**Note:** Touch screen devices do not have a hover state. See [No Hover](#) by Trent Walton for more information regarding this interesting usability quandary.

You can read more about the other types of pseudo selectors on the [sitepoint page](#).

## Universal Selector

Matches every element on the page.

For example, if we wanted **every element** to have a solid 1px wide border, we would use the following CSS rule:

```
* { border: 1px solid blue;}
```

- [See Demo 1](#)

For reasons that are likely obvious after the previous example, you should be careful with universal selectors. When might you want to use them?

The answer is, not often. But an example would be to **set the margins and padding for all elements** on the page to zero. *We'll learn a better way to do this shortly.*

```
* {
  margin: 0;
  padding: 0;
}
```

- [See Demo 2](#)

## Additional Reading

Here are some articles with more information about CSS Selectors.

- [CSS Selectors](#)
- [CSS contents and browser compatibility](#)

[Back to Class Four page »](#)