



Introduction

Next let's talk about the **descendant selectors**. A descendant selector is written with two or more selectors with white space between them:

```
ul li a {color: red}
```

When working with descendant selectors, you should read from right to left. So in this example we're actually looking for any a element that's inside a li element that's in turn is inside the ul.

In our example below, there are two a element in the HTML document. However, only one of them sits inside a

With the descendant selector, you should write it only as specific as you need it to be. There's no point in writing the full path to the element, when you could write it much more simply as something like

```
.nav a {}
```

This is more efficient but it's also more robust, because you might end up moving that nav to a different area of the page, and if the descendant selector is too specific, it may not work.

Descendant selectors are well supported across all modern browsers.

Exercise

We want to target the ul inside our navigational block:

```
.nav ul
{
  margin: 0;
  padding: 0;
}
```

Keep in mind that descendant selectors start from the right. In this example we're targeting any ul element that exist inside an element that has a class of nav.

Next up, we're going to style the lis inside the .nav:

```
.nav li
{
    display: inline;
    margin-right: 1em;
}
```

By setting them to `display: inline` we are making those elements sit beside each other in a row.

And the last descendant selector we're going to do is to target the `a` element inside the `nav`:

```
.nav li a { color: #fff; }
```