



CSS PART TWO

WEBSITE DEVELOPMENT I

```
<body>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci
  nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
</p>
<p>
  Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
  commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
  hendrerit turpis.
</p>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
  sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
  velit at orci.
</p>
<p>
  Lorem ipsum dolor sit amet,<span>consectetur adipiscing elit</span>. Cras
  sollicitudin, orci
  nec acilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
</p>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
  sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
  velit at orci.
</p>
</body>
```

How to style these paragraphs differently?

Just using p as the selector will set the style for them all.

```
p {
  color: black;
  background-color: teal;
}
```

CLASS SELECTOR

- We previously looked at elements selectors, which targets all elements (which translates to HTML tags) on a page. If we want to be more specific, class selectors is the next step. Instead of targeting all elements with a specific name, they target all elements that has a specific **class** name specified.
- A class selector looks just like an element selector, but instead of using names that are tied to the names of HTML elements, you make up the name and then you prefix it with a dot (.).

Using class to identify elements

```
<body>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci
  nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
</p>
<p class="withstyle">
  Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
  commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
  hendrerit turpis.
</p>
<p class="warning">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
  sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
  velit at orci.
</p>
<p>
  Lorem ipsum dolor sit amet,<span class="warning">consectetur adipiscing elit</span>.
  Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis
  enim velit at orci.
</p>
</body>
```

- To indicate that an element is a member of a class we use the **class** attribute.
- While the name of an element specifies its *type*, the class attribute lets you assign to it one or more *subtypes*.

```
<body>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci
  nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
</p>
<p class="withstyle">
  Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
  commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
  hendrerit turpis.
</p>
<p class="warning">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
  sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
  velit at orci.
</p>
<p>
  Lorem ipsum dolor sit amet,<span class="warning">consectetur adipiscing elit</span>.
  Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis
  enim velit at orci.
</p>
</body>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

USING CLASSES IN CSS

- Class names are referenced in CSS stylesheet as
 - element.classname

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

```
p {  
  background-color: white;  
  color: black;  
  font-family: times;  
  margin: 0.5em;  
  padding: 0.5em;  
}
```

```
p.withstyle {  
  background-color: olive;  
  color: navy;  
  font-family: sans-serif;  
  margin: 0.5em;  
  padding: 0.5em;  
}
```

```
p.warning {  
  background-color: yellow;  
  color: red;  
  font-weight: bold;  
}
```

CLASSES INDEPENDENT OF ELEMENTS

- May not have an element name preceding the period:

.classname

- Selector now matches *any* element of the given class

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, **consectetur adipiscing elit**. Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

```
p {  
  background-color: white;  
  color: black;  
  font-family: times;  
  margin: 0.5em;  
  padding: 0.5em;  
}
```

```
.withstyle {  
  background-color: olive;  
  color: navy;  
  font-family: sans-serif;  
  margin: 0.5em;  
  padding: 0.5em;  
}
```

```
.warning {  
  background-color: yellow;  
  color: red;  
  font-weight: bold;  
}
```

ID SELECTOR

- Now we will look at the most specific selector type: The **id** selector.
- The id selector is actually so specific that it only targets a single element on the page.
- An id selector looks just like a class selector, but instead of having a dot as the prefix, it uses the hash sign (#).
- It works just like classes, but instead of using a dot, we use a hash character, and instead of using the **class** attribute, we use the **id** attribute - **the difference lies in the fact that the id should be unique.** The value of the id attribute should be unique, according to the HTML specification, meaning that it can only be used on a single element per page.

ID SELECTOR

- Just like the class selector, you may limit an id selector to a specific element type by putting the name in front of the selector name, like this:

```
h1#main-header {  
    color: greenyellow;  
}
```

With this rule, this specific id selector will only apply to a header (h1) tag.

Unordered List Example

Fast food in Vermillion

- Burger King
- Dairy Queen
- Dominos
- Hardee's
- Pizza Hut

Ordered List Example

Fast food in Vermillion

1. Burger King
2. Dairy Queen
3. Dominos
4. Hardee's

DESCENDANT SELECTOR

- So far, we have only used selectors which directly targeted a specific element or element(s) with a specific id or class. Especially targeting elements of a specific type, e.g. all links or images, is very powerful, but what if you want to limit this, for instance to elements found only in a specific part of the page?
- A **Descendant selector**, allows you to limit the targeted elements to the ones who are descendants of another element. The syntax is very simple - you simply write the parent(s), separate with a space, and then the actual element you want to target.

DESCENDANT SELECTOR

```
<style>
p b {
    color: blue;
}
</style>
```

- In this example, I want all bold elements to be blue, but only if they are inside a paragraph tag.
- Using a descendant selector would easily allow you to change the colour of all links inside your main menu, without having to tag them all with a specific class!

DESCENDANT SELECTOR

A descendant doesn't need to be the direct child

- With this selector type, you should be aware that not only direct children are targeted - also children of the child (grandchildren) and so on will be targeted, all the way down through the hierarchy.

```
div.highlighted b {  
    color: blue;  
}
```

DESCENDANT SELECTOR

```
<div class="highlighted">  
  <b>Level 0...</b>  
  <div>  
    <b>Level 1...</b>  
    <div>  
      <b>Level 2...</b>  
    </div>  
  </div>  
</div>
```

Level 0...
Level 1...
Level 2...

Here, we target bold elements which are descendants of a div tag with the class "highlighted".

CHILD SELECTOR

- We have seen just how powerful the descendant selector can be, because it allows you to target ALL children and grandchildren (and so on) of one or several elements. However, sometimes this can be TOO powerful - sometimes you only want to target the direct children of an element. Fortunately for us, CSS has a selector for this as well!
- The syntax for using the direct child selector looks like this:

parent > child

CHILD SELECTOR

```
<style>
div.highlighted > b {
    color: blue;
}
</style>
```

- Now, only direct children of the parent are now affected.

CHILD SELECTOR

```
<style>
div.highlighted > b {
    color: blue;
}
</style>
```

- Now, only direct children of the parent are now affected.

Level 0...
Level 1...
Level 2...

SIBLING SELECTOR

- What if you want to target siblings instead? CSS has a couple of selector types for that as well, here we'll check out the general sibling selector.
- With the general sibling CSS selector, which takes a selector, followed by a tilde character (~) and then the selector you wish to target, you can target elements by requiring the presence of another element within the same parent element. Another requirement is that the first part of the selector needs to be present in the markup BEFORE the targeted element, even though they are all children of the same parent.

SIBLING SELECTOR

```
<style>
  h2 ~ p {
    font-style: italic;
  }
</style>
```

```
<h1>Hello, world!</h1>
<p>Some text here</p>
<h2>Hello, world!</h2>
<p>Some text here</p>
<p>More text here</p>
```

Hello, world!

Some text here

Hello, world!

Some text here

More text here

ADJACENT SIBLING SELECTOR

- We have just looked at the sibling selector, which allows us to select all elements which follows another element within the same parent. However, using the Adjacent sibling selector, you can limit this to only include the first element which comes directly after the first element in the markup.

ADJACENT SIBLING SELECTOR

<code><style></code>	<code><h1>Hello, world!</h1></code>
<code> h2 + p {</code>	<code><p>Some text here</p></code>
<code> font-style: italic;</code>	<code><h2>Hello, world!</h2></code>
<code> }</code>	<code><p>Some text here</p></code>
<code></style></code>	<code><p>More text here</p></code>
	<code><h2>Hello, world!</h2></code>
	<code><p>Text here as well...</p></code>
	<code><p>But no more!</p></code>

ADJACENT SIBLING SELECTOR

- With the adjacent sibling selector, we have just specified that the first paragraph element after all H2 elements should use italic text.
- The syntax for the adjacent sibling selector is - the two selector parts are simply joined by a plus character (+).

ADJACENT SIBLING SELECTOR

Hello, world!

Some text here

Hello, world!

Some text here

More text here

Hello, world!

Text here as well...

But no more!

- With the adjacent sibling selector, we have just specified that the first paragraph element after all H2 elements should use italic text.
- The syntax for the adjacent sibling selector is - the two selector parts are simply joined by a plus character (+).

STYLING LISTS

The list-style-type Property

- The list-style-type property defines the kind of marker that is to be associated with each item in the list.
- By default, an unordered list displays with an item marker of a bullet (disc). In nested unordered lists, the item marker changes to an open circle for the first level of indentation, and a square for the second level. What if you prefer to have the item marker be a square for the outermost list, a bullet for the next one, and an open circle for the third?

STYLING LISTS

```
ul {  
    list-style-type:square;  
}  
  
ul ul {  
    list-style-type:disc;  
}  
  
ul ul ul {  
    list-style-type:circle;  
}
```

- Vegetables
 - Turnips
 - Carrots
 - Garden Peas
- Fruit
 - Oranges
 - Apples
 - Golden Delicious
 - Granny Smith
 - Kiwis
 - Strawberries

STYLING LISTS

- For an ordered list you can change from the default numbering system to alphabetic characters or roman numerals, for example:

```
ol {  
  list-style-type:upper-roman;  
}
```

```
ol ol {  
  list-style-type:lower-roman;  
}
```

```
I. Kerry  
  i. Killarney  
  ii. Kenmare  
  iii. Dingle  
II. Cork  
  i. Youghal  
  ii. Baltimore
```

STYLING LISTS

- If you want no bullets or numbers:

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

STYLING LISTS

- Two other properties available to lists are:
 - list-style-position
 - list-style-image