



CSY1018

Web Programming

Topic 9

thomas.butler@northampton.ac.uk



Topic 9

- Advanced CSS selectors
 - *
 - +
 - ~
 - Element and class together
 - Attribute selector
 - first-letter
 - first-child
 - nth-child
 - Odd and even

Advanced CSS selectors

- CSS selectors are used to find elements on the page and target them for styling purposes
- We previously looked at several basic ones

```
h1 {  
  font-weight: bold;  
}
```

CSS selectors

- All CSS selectors can be *chained together* to make very complex expressions to find an element in the document
- **You can target any element on any page without using class (or id) attributes!**

```
main p a {  
  font-weight: bold;  
}
```

Basic selectors

- Previously we looked at basic selectors
- Tag Name e.g. *main* which targets all elements of a specific tag
- Class name e.g. *.className* which targets any element with the `class="className"`
- ID e.g. *#id* which targets any element with the `id="id"`
- Direct descendant operator e.g. *main > h1* which targets any h1 element directly inside a main element

Advanced selectors

- There are many other CSS selectors which can be used, though for most of these they are very much more situational and you will find yourself using them less frequently

Wildcard selector *

- The * selector can be used to target every element on the page
- This is quite often used as a *reset*. All paddings/margins can be turned off which means every element is treated the same. It is then up to you to add paddings and margins to elements that need them

```
<h1>Heading</h1>
```

```
<p>Paragraph</p>
```

```
<ul>
```

```
    <li>List item 1</li>
```

```
    <li>List item 2</li>
```

```
</ul>
```

Heading

Paragraph

- List item 1
- List item 2

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

Heading

Paragraph

List item 1

List item 2

+ selector

- The + selector can be used as a *combination selector* like the direct descendant operator
- It is used to target an element that *immediately follows another*

```
<h1>Heading</h1>
<p>Paragraph 1</p>
<ul>
  <li>List item 1</li>
  <li>List item 2</li>
</ul>
<p>Paragraph 2</p>
```

```
h1 + p {
  color: red;
}
```

Heading

Paragraph 1

- List item 1
- List item 2

Paragraph 2

~ selector

- The ~ selector is like the + operator but does not require elements to appear immediately after one another
- Only elements in the same parent tag are selected!

```
<h1>Heading</h1>
<p>Paragraph 1</p>
<ul>
  <li>List item 1</li>
  <li>List item 2</li>
</ul>
<p>Paragraph 2</p>
```

```
h1 ~ p {
  color: red;
}
```

Heading

Paragraph 1

- List item 1
- List item 2

Paragraph 2

Element and class combined

- You can combine element and class selectors together to perform a *logical and*
- The element is a h1 AND has a class of heading

```
<h1>Heading 1</h1>
```

```
<h1 class="heading">Heading 2</h1>
```

```
h1.heading {  
    color: red;  
}
```

Attribute selector

- The attribute selector can be used to select elements with a specific attribute and value
- This is very useful for styling different form elements differently

```
<form>
  <input type="text" />
  <input type="password" />
  <input type="submit" />
</form>
```

```
input {
  border: 2px solid red;
  background-color: darkred;
  color: white;
}
input[type="submit"] {
  border: 2px solid green;
  background-color: darkgreen;
}
```

Submit Query

:first-child

- The attribute selector can be used to select an element of a parent if it is the *first-child* element of that parent
- This is very useful for styling different form elements differently

```
<section>
  <h1>Heading</h1>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
</section>

<section>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
</section>
```

```
p:first-child {
  color: red;
}
```

Heading

Paragraph 1

Paragraph 2

Paragraph 1

Paragraph 2

:first-letter

- First-letter can target the first letter in an element
- This is useful if you want to style the first letter differently, e.g. like a novel

```
<section>
  <h1>Heading</h1>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
</section>
```

```
<section>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
</section>
```

```
p:first-letter {
  font-weight: bold;
  font-size: 2em;
}
```

Heading

Paragraph 1

Paragraph 2

Paragraph 1

:first-letter

- First letter and first child can be combined to target just the first letter of the first paragraph

```
<section>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  <p>Paragraph 3</p>
  <p>Paragraph 4</p>
</section>
```

```
p:first-child:first-letter {
  font-weight: bold;
  font-size: 2em;
}
```

Paragraph 1

Paragraph 2

Paragraph 3

Paragraph 4

:nth-child

- Nth-child can be used to select an element that is a given *index of its parent element*

```
<section>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  <p>Paragraph 3</p>
  <p>Paragraph 4</p>
</section>
```

```
section p:nth-child(3) {
  color: red;
}
```

Paragraph 1

Paragraph 2

Paragraph 3

Paragraph 4

:nth-child(odd)

- Nth-child odd can be used to target odd indexed elements
- This is useful for alternative row effects

```
<section>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  <p>Paragraph 3</p>
  <p>Paragraph 4</p>
</section>
```

```
section p:nth-child(odd) {
  background-color: #eee;
}
```

Paragraph 1

Paragraph 2

Paragraph 3

Paragraph 4

:nth-child(even)

- Nth-child even is the inverse of odd, both can be combined to give an alternating rows effect

```
<section>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  <p>Paragraph 3</p>
  <p>Paragraph 4</p>
</section>
```

```
section p:nth-child(odd) {
  background-color: #eee;
}

section p:nth-child(even) {
  background-color: #ccc;
}
```

Paragraph 1

Paragraph 2

Paragraph 3

Paragraph 4

Advanced selectors

- These selectors are used infrequently but allow you to solve complex problems
- It's useful to know they exist even if you do not remember the exact syntax

Advanced selectors

- This is not a complete list
- Take a look here: <https://css-tricks.com/almanac/selectors/> for a comprehensive list of selectors
- Note: Some of the selectors listed are *experimental* and may not work in all browsers

Exercises

- 1. Re-work last week's exercises so that the submit button *does not need a class name*
 - *Hint: use the attribute selector*
- 2. Download exercise2.zip and follow the instructions in the HTML file to add the correct styling
 - You will need to create the CSS file
 - Hint: You can use section:nth-child() to target a particular section, remember all selectors can be combined!
 - **DO NOT MODIFY THE HTML files! Your task is to set the element properties using only css! You do not need to add classes to elements which do not already have them.**