CS1115/CS5002

Web Development 1

Dr Derek Bridge

School of Computer Science & Information Technology
University College Cork

CSS selectors

- Revision:
  - A **CSS stylesheet** contains zero, one or more **rules**
  - A **rule** comprises:
    - one or more **selectors** and
    - a block of zero, one or more **declarations** in curly braces
- We'll split the selectors into
  - find selectors (which we cover in detail)
  - filter selectors (which we will mention briefly)

CSS: selectors that *find* elements

`*`	Matches all elements
`E`	Matches all elements with tag name `E`
`#I`	Matches the element with `id` of `I`
`.C`	Matches all elements with `class` of `C`
`E > F`	Matches all `F` elements that are children of `E` elements
`E F`	Matches all `F` elements that are descendants of `E` elements
`E + F`	Matches all `F` elements that are immediately preceded by sibling
`E ~ F`	Matches all `F` elements preceded by sibling `E`

The universal selector

- The **universal selector** * selects all elements:

```
* {
    color: red;
}
```

- More common (but not always identical) is to apply properties to the <body> element:

```
body {
    color: red;
}
```

Type selectors

- A selector can be a tag:

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

- We can have more than one selector, separated by commas:

```
h1, p, footer {
    color: red;
}
```

The `class` attribute in HTML

- In HTML, `class` is a global attribute
- You can invent its value. Several elements can share the same value, e.g.:

```
<ul id="recipe">
    <li class="ingredient">three <em>large</em> eggs</li>
    <li class="ingredient">two <em>pints</em> of <em>full fat</em> milk</li>
</ul>
```

Id selectors

- In CSS, you can use # to select by id:

```
#ingredients {
    color: red;
}
```

Class selectors

- In CSS, you can use . to select by class:

```
.cocktail {
    color: red;
}
```

Child and descendant selectors

- To select 'a' elements that are *children* of `li`s:

```
li > a {
    color: red;
}
```

- To select 'a' elements that are *descendants* of `nav` elements:

```
nav a {
    color: red;
}
```

Observation

- Use child and descendant selectors to avoid the novice's disease of *class-itis* (and *id-itis*)

Class exercise

- What do these select?

```
nav li li {
    color: red;
}

nav > li {
    color: red;
}
```

- What's the difference between these two rules?

```
section ol, ul {
    color: red;
}

section ol, section ul
{
    color: red;
}
```

- How would you select only `<h1>Ingredients</h1>`?
- How would you select only the ingredient `li`'s?

Sibling selectors

- Based on immediately preceding sibling
  - To select a `p` whose immediately preceding sibling is an `h1`:

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

  - To select an `li` whose immediately preceding sibling is another `li`:

```
li + li {
    color: red;
}
```

- Based on any preceding sibling
  - To select a `p` any of whose preceding siblings is an `h1`:

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

Class exercise: what do these select?

```
<ul>
    <li id="pastry">For the pastry:
        <ul>
            <li class="ingredient">flour</li>
            <li class="ingredient">butter</li>
        </ul>
    </li>
    <li id="filling">For the filling:
        <ul>
            <li class="ingredient">stewing beef</li>
            <li class="ingredient">an onion</li>
            <li class="ingredient">Guiness</li>
        </ul>
    </li>
</ul>
```

1. `li`
2. `.ingredient`
3. `#filling li`
4. `li li`
5. `.ingredient + .ingredient`