CSS

COMP1004 WEB TECHNOLOGIES

CSS - CASCADING STYLE SHEETS

- Adds presentational Styles to our HTML
- Standardised; currently on version 3
- Rules target HTML and set properties
- Lots of style properties
- Selectors are used to target the HTML content to be styled.
- Parsed and rendered by Browsers
- CSS is in a separate document to your HTML and connected via a link tag

<link rel=”stylesheet” type=”text/css” href=”styles.css” />

CSS - EXAMPLE

```
table {
      width:500px;
      background-color:#0e0e0e;
}

table a{
      text-decoration:none;
      font-size:10px;
      color: green;
}

#my-link{
      display:block;
}
```

- Blue part is the **Selector**
- Black part is the **Property**
- Orange part is the **Value**
- Rules are separated by the **Curly brackets** (braces or squiggly brackets)
- Properties and values are separated by a **colon**
- Each property-value pair is separated ended by a **semicolon**

CASCADING

- Styles (CSS Rules) can be declared in a number of places
- These rules cascade down to give an element its final style:
- There is a hierarchy of style sources
  - The Browser default styles
  - The User's custom styles
  - The Page Authors styles
- The Page Authors styles will have priority.

CASCADING

- The basic **browser** styles are the default values for each CSS property as
- defined by the CSS specification.
  - Not all browsers are the same however.
- Custom User styles will then **overwrite** or add to these rules
- **Page author (you)** styles come next in 3 levels
  - **An external file**. The order styles are applied is based on the order linked to the HTML document
  - Inside the HTML document. **A <style> tag** can be used **(Don't do this)**
  - On a HTML Element. **A style attribute** can be used **(Don't do this)**
- A style rule can be overwritten many times

CASCADING - OVERWRITING

- Rules that are declared after another for the same selector will overwrite the previous matching properties.
- Here the background is set to red & the font size to 22 pixels.
- Then it is overwritten so the background is green.
- The font size will stay at 22 pixels as this has not been overwritten.

```
p {
    background-color:red;
    font-size:22px;
}

p {
    background-color:green;
}

p {
    background-color:purple;
    background-color:green;
}
```

CASCADING - OVERWRITING

- Some properties will be inherited from a parent element
- For example when setting the color of a paragraph, a strong tag nested inside will also receive that style.

```
p {
        color:red;
}
p {
        color:blue;
}
<p>Hello my name is <strong>James</strong>. Nice to meet you.</p>
```

SELECTORS

SELECTORS

- CSS is more complex than HTML; More moving parts
- Selectors are an expression used to make a selection from the HTML on your page
- Selectors can make use of certain HTML attributes to better target your HTML content
- Number of different types of selector
  - Type Selector <a href="www..
  - Class Selector <a class="special-link" href..
  - ID Selector <a id="my-link" href..
  - Pseudo Selector <a href..
  - Descendant Selector <p><a href..

```
a { property:value;}
.special-link { property:value;}
#my-link { property:value;}
a:hover { property:value;}
p a { property:value;}
```

TYPE SELECTORS

- Type selectors select all the elements of the same type
- Most basic type of selector and can be used without any additional markup
- Very broad selector

```
<p>Hello world</p>
p {
      background-color:red;
}
```

CLASS SELECTORS

- Class selectors select all the elements which have a given class as the value of their class attribute
- CSS Classes are designed to be used more once
- Build your own UI “Components” using classes and nested elements
- Identified in the CSS by a “.” before a unique name

```
<p class=”first-para”>Hello world</p>
```

```
.first-para{
      background-color:red;
}
```

ID SELECTORS

- ID selectors select the element which has a given id as the value of its id attribute
- ID selectors are designed to be used only one element.
- Identified in the CSS by a “#” before a unique name

```
<p id=”my-para”>Hello world</p>
```

```
#my-para{
      background-color:red;
}
```

DESCENDANT SELECTORS

- Descendant Selectors target a specific nesting
- “All spans within a paragraph”
- Can be chained to many levels of depth

```
<p>Hello <span>world</span></p>


p span{
      background-color:red;
}
```

CHILD SELECTORS

- The child selector targets elements that are a direct child of a given selector/element
- Only 1 level of depth vs descendant selector

```
<ul>
  <li>List item</li>
</ul>
```

```
ul>li{
      background-color:red;
}
```

UNIVERSAL SELECTOR

- Will select all elements within the current context
- Excellent for quickly removing all borders/margin/padding from any elements

```
<p>Hello world</p>
<h1>howdy</h1>


* {
      font-size:16px;
}
```

PSEUDO-CLASS SELECTORS

- Pseudo-Class selectors target an element at a specific state.
- Most common use is changing a style when hovering. (mouse pointer is over something)
- Another is selecting the first child element

```
<a href=”..” title=””>Hello</a>
a {
      background-color:green;
}
a:hover {
      background-color:red;
}
```

```
<div>
  <p>first child</p>
  <p>second child</p>
</div>

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

PSEUDO-ELEMENT SELECTORS

- Pseudo-Element selectors target a specific part of an element's content
- First line, First Letter, “Before”, “After”

```
<a href=”..” title=””>Hello</a>
a {
      font-size:16px;
}
a:first-letter {
      font-size:20px;
}
```

SPECIFICITY

SPECIFICITY

- In addition to cascading rules
- “Last” declaration “wins”
- Selectors form a hierarchy
  - Inline (style attribute)
  - ID
  - Class
  - Element
- Combining selectors leads to more specific selection, more important rule.

```
p {
      background:red;
}

.class-para {
      background:blue;
}

#id-para {
      background:green;
}

<p class=”para” id=”id-para”>Which colour?</p>
```

EXAMPLES

W3Schools selector examples

http://www.w3schools.com/cssref/trysel.asp

CSS Diner

http://flukeout.github.io/