ELT Computer Science

Sheriffo Ceesay
Nnamdi Ekwe-Ekwe

Greatest appreciation goes to Martin and Shyam for sharing this well formatted lecture materials.

Any questions?

- How was the lab session?
- What did you produce?
- Is there anything you'd like to try?

CSS selectors

What if we don’t want to affect **every instance** of a tag?

- just the first h1?
- just these three paragraphs?

We can apply CSS to specific **id**s and **class**es

- *<h1 id=”main_header”>...</h1>*
- * ... *
- You can give several classes to a single element
  - *<p class=”important_para intro”>... <!-- assign class “important_para” **and** “intro” -->*

Actual text of id or class doesn’t matter

- just has to be unique

Referring to elements

- To refer to an **id**, use a hash **#**

```
#main_header {
        color:blue;
}
```

- To refer to a **class**, use a dot **.**

```
.important_para {
        background-color:orange;
}
```

Combining selectors

You can combine selectors in various ways

```
h1#main_header {
    color:red;
}
```

- Only affects <*h1*>s with *id=”main_header”*

```
#main_header.important_para {
    color:blue;
}
```

- Only affects elements with *id=”main_header”* **and** *class=”important_para”*

Multiple selectors

Duplicating rules can be necessary

- multiple ways to describe relevant tags
- e.g. multiple <*p*>s need the same colour

Could just copy/paste rules

- but this is hard to read!

Solution: use a comma **,**

```
h1, h2.important, #header p {
    font-weight:bold;
}
```

- applies to all elements matching any selector

Nesting selectors

You can select tags within other tags

```
ol li {
    font-style:italic;
}
```

- Only affects <*li*>s anywhere **within** an <*ol*>
  - even if they're inside something inside the <*ol*>

```
body>h1 {
    text-decoration:underline;
}
```

- Only affects <*h1*>s **immediately** within the <*body*>

More selectors

You can be very specific

- choose to apply only at certain times
- or invert your choices

#header a:hover { ... }

- links inside the #header **while** mouse is over them

h1 .important:not(p:first-child) { ... }

- what does this affect?

There are always more...

Two new tags

Most tags mean something

- <*h1*> is a header: big text
- <*em*> is emphasised: in italic

What if we want nothing?

- element just exists to be given custom rules
  - via various selectors

Two tags exist

- <*div*> - has a new line before and after (normally)
- <*span*> - has absolutely nothing

More CSS features

- CSS offers you control
  - over pretty much anything
  - e.g. size, colour
- One important thing: **positioning**
  - where should everything be?
  - relative to what?
- Simple example: centering text

```
h1 {
    text-align:center;
}
```

- text is at the centre of the element

CSS positioning: relative

Given where an element is by default

- let's just move it a little

We need to specify **position** type

- and the actual position!

```
h1 {
        position:relative;
        top:15px; left:100px;
}
```

- This moves the <*h1*> from where it **would** be otherwise

CSS positioning: absolute

We can also force a specific position

- not relative to some other element

```
h1 {
        position:absolute;
        left:150px;
        top:100px;
}
```

- Position is relative to the **top left** of the web page
- Element is “removed from flow”
  - Other elements can overlap it!

CSS positioning: fixed

We can place something on the **screen** not the page

```
h1 {
    position:fixed;
    left:150px;
    top:100px;
    height:50%;          /* 50% of screen height, not page height */
}
```

- This won't move if page is scrolled

CSS positioning: float

What if we just want one thing next to another?

- just to the right
- just to the left

```
.go-left {
        float:left;
}
```

- all elements with *class="go-left"* will move towards the left
  - as far as they can

CSS positioning: clearing floats

An element with *float:* ceases to be **block level**

- this means no new line before or after
- even for <*div*>, <*h1*> etc.!

After *float:*ed elements, put a *clear:both* one:

```
<div class='go-left'>Your contents here</div>
<div class='clear-float'></div> <!-- note: no need for any contents -->

div.clear-float:
    clear:both;
}
```

More CSS

CSS is extensive
- but must be learned by practice

We can’t cover everything
- even if we spent the semester on CSS!

Try things out!

If you have specific questions:
- ask me
- ask Google