HTML and CSS

- This section covers general overview of HTML and CSS as preparation for the next section (I'll put the name here once I know it)

HTML

Elements

An element consists of a start tag, content, and an end tag. Here's an example:

```
<li>three <em>large</em> eggs</li>
```

Some elements (**void** elements) only have start tags and have no end tags or content. Examples include:

- `<input>`
- `<br>`
- `<hr>`

Heirarchy

The elements altogether form a **tree**, where each element is a **node**.

Leaf elements are those with no **children**, i.e. no elements nested within them.

Ancestor and **descendant** are generalised versions of parent and child, to include further generations.

Attributes and Values

Start tags can contain attributes, which are given values, e.g. for `<img>`:

```
src="URL_of_image"
alt="Text to be displayed if the image can't be"
title="Text to be displayed when the user hovers over the image"
```

The ID Attribute

This attribute can be applied to any element. It can have any value you like, but all ids in your document must be unique.

IDs are used for 3 things:

- For unique selectors in CSS.
- Providing anchors within the webpage (links that jump around within a single webpage).
- For labels in forms.

The Class Attribute

This attribute can also be applied to any element. Unlike id, multiple elements can have the same class.

Class is for formatting with CSS.

CSS

A CSS Stylesheet consists of one or more **rules.**

A rule comprises:

- one or more **selectors**
- and one or more **declarations**

Example:

```
h1 {                            /* h1 is a selector */
    color:red;
    background-color: blue;         /* background-color:blue; is a
declaration */
    }
h2 {
    color: black;
    background-color: black;
    }
```

CSS can be included as an **external style sheet**, an **internal style sheet**, or as **inline styles.**

External stylesheets are preferred because it's easier to maintain. All pages in one site link to the one stylesheet, so you can control the look of the whole website from one place.

Selectors Which Find Elements

Selector	What It Matches
*	All elements
E	All elements with tag name 'E'
#I	All elements with id 'I'
.C	All elements with class 'C'
E > F	All F elements that are children of E elements
E F	All F elements that are descendants of E elements
E + F	All F elements that are are immediately preceeded by a sibling E element
E ~ F	All F elements preceeded by a sibling E element

Selectors Which *Filter*

- Attribute Selectors:
  E[foo] finds E elements that have an attribute foo.
- Pseudo-classes (after a colon), e.g.:
  E:first-child finds E elements which are the first child of their parent element
- Pseudo-elements (after two colons), e.g.:
  p::first-letter finds the first letter of each p element