Entities

- **Special Characters**
  - Symbols such as +, -, %, and & are used frequently.
  - Not all Web browsers display these symbols correctly.
  - HTML uses a little computer shorthand to tell the browser how to interpret these symbols.
  - [www.w3.org/TR/REC-html40/sgml/entities.html#h-24.2.1](http://www.w3.org/TR/REC-html40/sgml/entities.html#h-24.2.1) - Contains a complete list of the characters supported by HTML.
## Entities

<table>
<thead>
<tr>
<th>Char</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>&amp;</td>
<td>ampersand</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>less than</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
<td>greater than</td>
</tr>
<tr>
<td>©</td>
<td>©</td>
<td>copyright</td>
</tr>
<tr>
<td>®</td>
<td>®</td>
<td>registered trademark</td>
</tr>
<tr>
<td>²</td>
<td>²</td>
<td>superscript²</td>
</tr>
<tr>
<td>³</td>
<td>³</td>
<td>superscript³</td>
</tr>
<tr>
<td>/</td>
<td>´</td>
<td>acute accent/fada</td>
</tr>
<tr>
<td>\</td>
<td>`</td>
<td>grave accent</td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td>hash sign</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>percent sign</td>
</tr>
</tbody>
</table>
Entities

Special Characters

- Probably the most important of these special characters is the non-breaking space: 

  &nbsp;

- Used to insert a space inside a HTML document.

- Also used to create an empty cell in a table.
Even More Tags

- **Meta Tags**
  - Contained within the `<head>` tag.
  - Can use as many `<meta>` tags as you want in your page.
  - Doesn’t appear in the document.

- **Used to**
  - Specify the `charset`
  - Identify the page’s author,
  - Identify keywords used for searching,
  - A brief description to appear in search results
  - Give commands to the browser
Even More Tags

- **Meta Tags**

- Improve Searching

  - Search Engines add the content of your Web pages to their indexes

  - When a potential visitor enters a search phrase, the search engine checks its index to find that phrase and returns any pages that include it.
Even More Tags

- **Meta Tags**

- **Improve Searching**
  - You can use the `<meta>` tag to include product names, geographic locations, industry terms, and synonyms.
  - There are three `<meta>` tags that work to help improve your chances of being found by a search engine
    - Keywords
    - Description
    - Author
Even More Tags

- **Meta Tags**

- Improve Searching
  - *Keywords* – words that you feel people might use to search for your web page, or synonyms for words in your document.

  - *Description* – usually a paragraph of information about your page. Some search engines use this description to describe your page; other search engines use the first few lines of text in your document.

  - *Author* – This is your opportunity to shine. Just in case someone is searching for your name, they will find it if you enter that information into the `<meta>` tag.
Even More Tags

- **Meta Tags**
  - **Improve Searching**
    - Meta information for search engines comes in pairs: **name** and **contents**.

```html
<html>
<head><title>Your HTML Page</title>
<meta name="keywords" contents="keywords, that people, might, use, to, search, for, your, page" />
<meta name="description" contents="a brief paragraph describing your document" />
<meta name="author" contents="your name" />
</head>
<body>Insert your content here</body>
</html>
```
Even More Tags

- **Meta Tags**

- **Refresh and Redirect**
  - Replace one page with another
  - Redirect a link
  - Include a splash page (please don’t)
  - You can use meta information to force the page to change within a given time span.
  - Can use the refresh tag to refresh the same page – use if there are frequent updates

  <meta http-equiv="refresh" content="time in seconds; URL=URL for new page" />
CSC7026

Cascading Style Sheets
Applying CSS

- There are three ways of applying CSS to HTML:
  - Inline
  - Internal
  - External
Inline Styles

- **Inline** styles are put straight into the HTML tags using the `style` attribute.

- They look something like this:
  ```html
  <p style="color: red">text</p>
  ```
  This will make that specific paragraph red.

- But remember - the ideal is that the HTML should be a stand-alone, *presentation free* document, and so in-line styles should be avoided wherever possible.
Internal Styles

- **Internal** styles are used for the whole page. Inside the head tags, the **style** tag surrounds all of the styles for the page.

```html
<html>
<head><title>CSS Example</title>
<style type="text/css">
  p {
    color: red;
  }

  a {
    color: blue;
  }
</style> ...
</head>
```
Internal Styles

- This will make all of the paragraphs in the page red and all of the links blue.

- Similarly to the inline styles, you should keep the HTML and the CSS files separate, and so we are left with...
External Styles

- **External** styles are used for the whole, multiple-page website. There is a separate **CSS file**, which will simply look something like this:

```css
p {
    color: red;
}

a {
    color: blue;
}
```
External Styles

If this file is saved as `mystyle.css' (and it is in the same folder as the html file) then it can be linked to in the HTML like this:

```html
<html>
<head> <title>CSS Example</title>  
<link rel="stylesheet" type="text/css" href="mystyle.css" />
...  
```
 Whereas HTML has tags, CSS has 'selectors'. Selectors are the names given to styles in internal and external style sheets.

Selectors are simply the names of HTML tags and are used to change the style of a specific tag.

For each selector there are 'properties' inside curly brackets, which simply take the form of words such as color, font-weight or background-color.

A value is given to the property following a colon (NOT an 'equal' sign).
Syntax

- Each property:value pair is known as a **declaration**.

- **Semi-colons** separate the declarations.

  ![Diagram of selector and declarations]

  ```
  h1 {
    color: blue; font-size: 12px;
  }
  ```

- Do not leave spaces between the property value and the units! "margin-left:20 px" (instead of "margin-left:20px") will work in IE, but not in Firefox or Opera.
Syntax

body {
  font-size: 0.8em;
  color: navy;
}

- This will apply the given values to the `font-size` and `color` properties to the `body` selector.

- So basically, when this is applied to an HTML document, text between the body tags (which is the content of the whole window) will be 0.8 ems in size and navy in colour.
There are many property-specific units for values used in CSS, but there are some general units that are used in a number of properties and it is worth familiarising yourself with these before continuing.

- 'em' (such as font-size: 2em) is an element approximately equal to the **height of a character**.
- 'px' (such as font-size: 12px) is the unit for **pixels**.
- 'pt' (such as font-size: 12pt) is the unit for **points**.
- '%' (such as font-size: 80%) is the unit for **percentages**.

Lengths and Percentages

- Other units include 'pc' (picas – 1/6 of an inch), 'cm' (centimetres), 'mm' (millimetres) and 'in' (inches).

- When a value is zero, you do not need to state a unit. For example, if you wanted to specify no border, it would be `border: 0`. 
Lengths and Percentages

- **Note:** A web page is not a static, absolute medium. It is meant to be flexible and the user should be allowed to view the web page how they like, which includes the font size AND the size of the screen.

- Because of this, it is generally accepted that 'em' or '%' are the best units to use for font-sizes, rather than 'px', which leads to non-resizable text in most browsers, and should be used sparingly, for border sizes for example.
CSS brings **16,777,216** colours to your disposal. They can take the form of a **name**, an '**rgb**' (red/green/blue) value or a '**hex**' code.

CSS3 supports **HSL** values.

**red**
is the same as
**rgb** (255,0,0)
which is the same as
**rgb** (100%,0%,0%)
which is the same as
**#ff0000**
which is the same as
**#ff0000**
and
**hsl** (0,100%,50%)
Colours

- All modern browsers support 140 color names: http://www.w3schools.com/colors/colors_names.asp

- The 3 values in the rgb value are from 0 to 255, 0 being the lowest level (e.g. no red), 255 being the highest level (e.g. full red). The values can also be a percentage.

- Hexadecimal is a base-16 number system. We are generally used to the decimal number system (base-10, from 0 to 9), but hexadecimal has 16 digits, from 0-9 and a-f.
Colours

- The hex number is defined by a hash character (#) and can be three or six digits in length.

- Basically, the three-digit version is a compressed version of the six-digit (#f00 becomes #ff0000, #c96 becomes #cc9966 etc.).

- The three-digit version is easier to decipher (the first digit, like the first value in rgb, is red, the second green and the third blue) but the six-digit version gives you more control over the exact colour.
'color' and 'background-color'

- Colours can be applied by using `color` and `background-color` *(note that this must be the American-English 'color' and not 'colour').*

- A blue background and yellow text could look like this:

```css
h1 {
  color: yellow;
  background-color: blue;
}
```

- These colours might be a little too harsh, so you could change the code of your CSS file for slightly different shades:
'color' and 'background-color' properties can be applied to most HTML elements, including body, which will change the colours of the page and everything in it.

```html
body {
    font-size: 0.8em;
    color: navy;
}

h1 {
    color: #ffc;
    background-color: #009;
}
```
You can alter the size and shape of the text on a web page with a range of properties, outlined below:

- font-family
- font-size
- font-weight
- font-style
- text-decoration
- text-transform
- Text Spacing
This is the font itself, such as 'Times New Roman', 'Arial' or 'Verdana'.

The font you specify this way must be on the user's computer, so there is little point in using obscure fonts. There are a select few 'safe' fonts (the most commonly used are arial, verdana and times new roman).

But you can specify more than one font, separated by commas. The purpose of this is that if the user does not have the first font you specified, the browser will go through the list until it finds one it does have. This is useful because different computers sometimes have different fonts installed.
So **font-family: arial, helvetica**, for example, is used so that similar fonts are used on PC (which usually has arial, but not helvetica) and Mac (which may not have arial, and so helvetica, which it does normally have, will be used).

Note: if the name of a font is more than one word, it should be put in quotation marks, such as **font-family: "Times New Roman"**.
font-size

- The size of the font.

- Be careful with this - text such as headings should not just be a paragraph in a large font; you should still use headings (h1, h2 etc.) even though, in practice, you could make the font-size of a paragraph larger than that of a heading (not recommended for sensible people).
font-weight

- This states whether the text is **bold** or not.

- The most common values are `font-weight: bold` or `font-weight: normal`.

- In theory it can also be **bolder**, **lighter**, **100**, **200**, **300**, **400**, **500**, **600**, **700**, **800** or **900**, but some browsers aren’t so sure about this malarkey so it's safer to stick with **bold** and **normal**.
font-style

- This states whether the text is *italic* or not.

- It can be **font-style: italic** or **font-style: normal**.
text-decoration

- This states whether the text is underlined or not. This can be:
  - `text-decoration: overline`, which places a line above the text.
  - `text-decoration: line-through`, strike-through, which puts a line through the text.
  - `text-decoration: underline` SHOULD ONLY BE USED FOR LINKS because users generally expect underlined text to be links.

- This property is usually used to decorate links, such as specifying no underline with `text-decoration: none`.
text-transform

- This will change the case of the text.
  - `text-transform: capitalize` turns the first letter of every word into uppercase.
  - `text-transform: uppercase` turns everything into uppercase.
  - `text-transform: lowercase` turns everything into lowercase.
  - `text-transform: none` I'll leave for you to work out.
body {
    font-family: arial, helvetica, serif;
    font-size: 0.8em;
}

h1 {
    font-size: 2em;
}

h2 {
    font-size: 1.5em;
}

a {
    text-decoration: none;
}

strong {
    font-style: italic; text-transform: uppercase;
}
Text Spacing

- The **letter-spacing** and **word-spacing** properties are for spacing between letters or words. The value can be a length or **normal**.

- The **line-height** property sets the height of the lines in an element, such as a paragraph, without adjusting the size of the font. It can be a length, a percentage or **normal**.
Text Spacing

- The `text-align` property will align the text inside an element to left, right, center or justify.

- The `text-indent` property will indent the first line of a paragraph to a given length or percentage. This is a format usually used in print, and rarely in digital media such as the web.
Text Spacing

```css
p {
  letter-spacing: 0.5em;
  word-spacing: 2em;
  line-height: 1.5em;
  text-align: center;
}
```
Margins and Padding

- **margin** and **padding** are the two most commonly used properties for spacing-out elements. A margin is the space **outside** of the element, whereas padding is the space **inside** the element.

```css
h2 {
  font-size: 1.5em;
  background-color: #ccc;
  margin: 1em;
  padding: 3em;
}
```
Margins and Padding

- You will see that this leaves one-character width space around the secondary header and the header itself is fat from the three-character width padding.

- The four sides of an element can also be set individually. margin-top, margin-right, margin-bottom, margin-left, padding-top, padding-right, padding-bottom and padding-left are the self-explanatory properties you can use.
The Box Model

- Margins, padding and borders (later) are all part of what's known as the **Box Model**. In the middle you have the element box (let's say an image), surrounding that you have the padding box, surrounding that you have the border box and surrounding that you have the margin box. It can be visually represented like this:

![Diagram of the Box Model]

- The box model can be applied to every element on the page.
Borders can be applied to most HTML elements within the body.

To make a border around an element, all you need is `border-style`. The values can be `solid`, `dotted`, `dashed`, `double`, `groove`, `ridge`, `inset` and `outset`.

`border-width` sets the width of the border, which is usually in pixels. There are also properties for `border-top-width`, `border-right-width`, `border-bottom-width` and `border-left-width`.

Finally, `border-color` sets the colour.
Borders

```html
h2 {
    border-style: dashed;
    border-width: 3px;
    border-left-width: 10px;
    border-right-width: 10px;
    border-color: red;
}
```

- This will make a red dashed border around all HTML secondary headers (the `h2` element) that is 3 pixels wide on the top and bottom and 10 pixels wide on the left and right (these having over-ridden the 3 pixel wide width of the entire border).
You should download the HTML file `cssintro.htm` and add the line linking the HTML file to the CSS file.

```html
<link rel="stylesheet" type="text/css" href="mystyle.css" />
```

The code that follows covers all of the CSS methods in this lecture. If you copy and paste this into notepad and save it as your CSS file and look at the HTML file then you should now understand what each CSS property does and how to apply them.

The best way to fully understand all of this is to mess around with the HTML and the CSS files and see what happens when you change things.
CSS Code

body {
    font-family: arial, helvetica, sans-serif;
    font-size: 80%;
    color: black;
    background-color: #ffe;
    margin: 1em;
    padding: 0;
}

/* By the way, this is a comment */
p {
    line-height: 1.5em;
}
h1 {
    color: #ffc;
    background-color: #900;
    font-size: 2em;
    margin: 0;
    margin-bottom: 0.5em;
    padding: 0.25em;
    font-style: italic;
    text-align: center;
    letter-spacing: 0.5em;
    border-bottom-style: solid;
    border-bottom-width: 0.5em;
    border-bottom-color: #c00;
}

CSS Code (cont.)
CSS Code (cont.)

h2 {
    color: white;
    background-color: #090;
    font-size: 1.5em;
    margin: 0;
    padding: 0.1em;
    padding-left: 1em;
}

h3 {
    color: #999;
    font-size: 1.25em;
}
CSS Code (cont.)

```css
img {
  border-style: dashed;
  border-width: 2px;
  border-color: #ccc;
}

a {
  text-decoration: none;
}

strong {
  font-style: italic;
  text-transform: uppercase;
}
```
li {
  color: #900;
  font-style: italic;
}

table {
  background-color: #ccc;
}