Lab 1: HTML

Objectives

1. Ensure that webspace on *public_html* is working and understand how to publish to *homepages.shu.ac.uk*.
2. Review editing options.
3. Basic HTML structural, block and inline elements.
4. Add hyperlinks between pages and insert images.

Setting up your webspace

Our first simple pages with HTML and CSS do not require to be published to a webserver and can be tested locally.

If you would like to test your pages through a web server you should save them to the f:/public_html folder.

To test pages locally you can however save them to any directory you have access to.

Structure your files

Good web developers take time to structure their sites sensibly. It would be useful to create a folder for each week of the module as we will be creating content in all of the labs. Starting with a good naming convention can also help. Generally avoiding mix cases names and working in lowercase is tidy. Also avoid the use of special characters in file and folder names. In particular don't use white space in file or folder names as this will be translated by a browser into *%20*.

Under f:\public_html create a folder structure as follows:

```
f:\public_html\year1\ppd\lab1\
```

Most of the projects we build will include images, style sheets and scripts. It is a good practice to create folders in your site for this kind of content for example an *images* folder for all your JPGs, PNGs and GIFs, a *scripts* folder for your Javascript and a *styles* folder for your CSS.

Extract the contents of the ZIP file on Blackboard for this lab. Unzip the contents such that your file structure appears as:

```
f:\public_html\year1\ppd\lab1\
f:\public_html\year1\ppd\lab1\images
f:\public_html\year1\ppd\lab1\filltext
```

HTML Editors - The Choice is Yours

To edit HTML and CSS file all you need is a text editor. These can be sophisticated IDE (Integrated Development Environments) like Visual Studio, freeware like Sublime Text or even basic text editors like Notepad. It is recommended that you experiment with different editors to see which suit your workflow. Whichever tool you use it should help you, not hinder you, in your coding. Modern text editors will provide help with syntax and code checking so it is worth investing time to learn the tricks associated with your chosen tool.

For the purposes of this lab we'll be using Visual Studio Code a light weight version of Visual Studio. It is available via Apps Anywhere.

Setting Up Visual Studio Code

To open the files for the lab in Visual Studio code simply drag the folder created above into the application. The files will now appear in the side bar. This can be to toggled on and off with `Ctrl B`.

> This is an important step to ensure that you keep a track on the files in your website and saves a lot of problems related to editing the incorrect files.

Creating a file

To create a new file, use the new file icon in the side bar. Create a file called index.html ensuring your add the .html file extenstion.

HTML Structure

HTML is a mark-up language with 'tags' identifying parts of the document to behave in a certain way, for example, to be a link or to make the text bold. Tags generally consist of an opening and closing pair. They are written in lower case using angled brackets ie `<html>`.

Our first HTML tag (or element) will open and close the entire document.

Open *index.html*. It will be blank. Add the following:

```
<html>
</html>
```

Tip: With Visual Studio Code you can simply type html and then hit tab to add the above.

Inside of the `<html>` element go all the other HTML elements required to create our page. A HTML document consists of a head and body. The head is where information about the file is placed, the body is where the visual content of the page is placed. Unless otherwise stated the bulk of HTML element are place inside the `<body>` tag. A HTML file should only have one `<head>` and one `<body>` element.

Extend your code as follows:

```
<html>
      <head>
      </head>
      <body>
      </body>
</html>
```

HTML has a strict hierarchy that dictates which tags can be placed inside of which. As such the `<html>` element is the parent of all the other elements in HTML.

Tip: It may help you to indent tags as in the above example. White spacing is ignored by the web browser so feel free to use as much as you like to make your documents more readable.

Doctype and Meta Tags

The above is a very simplified version of the basic HTML file. In reality you will need to provide additional information for the browser about the type of HTML your file contains and the character set and language you intend to use.

Extend your code in *index.html* as follows:

```
<!doctype html>
<html>
     <head>
     <meta charset="UTF-8">
     </head>
     <body>
     </body>
</html>
```

The `<!doctype html>` line should always appear at the top of the file to indicate the file is a HTML5 document.

The `<meta>` tag above indicates the charset encoding that will be used by the file.

Tip: Meta tags can be used for provide a wide range of information about the file - some uses include search engines, social media and mobile compatibility.

Tip: In Visual Studio Code when creating a new file which require the HTML skeleton code you can just type `!` and then tab.

Page Title

To add a page title, that is seen at the top of the browser window, we use the `<title>` tag and this is a child of the `<head>` tag.

Add the following to index.html:

```
<head>
     <title>My First Web page</title>
</head>
```

It is important that you add the title because:

- Users can read it.
- It will appear in a browsers tab.
- It is the default name used when users bookmark your page.
- It will help the user identify the page when using the Back button.
- It is one of the many things referenced by search engines to find your page.

You may consider using a naming convention that will give your users a sense of where they are within your site by using a separating character such as the 'pipe' ie `Sheffield Winter Gardens | News | New Stall to Open`.

How to View Your Pages Locally

As you make additions to your webpage you should view them in a browser. In Visual Studio code select the file in the side panel and right click to find `Copy Path`. You can then paste this in the address bar of your Browser.

For debugging purposes, you would be advised to open the file in Google Chrome. Chrome has a range of sophisticated tools to help in web page development - collectively these are referred to as the Chrome Console.

As you amend your HTML pages in Visual Studio Code, simply save it and then in Google Chrome refresh.

How to View Your Pages Via homepages.shu.ac.uk

To view your pages through a webserver. You need to have saved your files to *f:/public_html*.

If you saved your files to:

```
f:\public_html\year1\ppd\lab1\index.html
```

You can view the pages through:

```
homepages.shu.ac.uk/~<STUDENT-NUMBER>/ppd/lab1/index.html
```

Notice the tilde (~). Put your student ID after it.

As you edit and save your page, to preview the changes simply refresh the page.

When pages are viewed through this *http* route they can be viewed from outside SHU. However, this feature is disabled by default.

If you wish to view your site from outside of SHU on the homepages.shu.ac.uk server you need to use the Managed Desktop Assistance Centre. This is found by typing MD in the start menu. Use the Self Service Icon and select `Manage Public_HTML Folder`. This may take a second or two to load but will give you the option to `Make Web Page Viewable from Outside SHU`.

Tip: Watch the Video https://youtu.be/GwM7Vj7ES08 to see how this works.

Block Elements

HTML tags fall into different categories. Block elements are those that by default force a new line of content before and after them. The most common block element is `<p>` used to define paragraphs.

When experimenting with the following code snippets ensure you place them in the index.html file. They should also all go inside of the `<body>`.

```
<body>
<p>This is a paragraph of text.</p>
</body>
```

HTML provides six tags `<h1>` through to `<h6>` that provide a heading hierarchy for a document. `<h1>` is the most important heading through to `<h6>` as the least important of the six.

```
<h1>This is heading style one</h1>
<h2>This is heading style two</h2>
<h3>This is heading style one</h3>
<h4>This is heading style two</h4>
<h5>This is heading style one</h5>
<h6>This is heading style two</h6>
```

By default the heading styles appear bold and decrease in size with `<h6>` the smallest. Note that the default styling of these elements can changed with CSS Cascading Style sheets. That is colours, fonts and sizing can be added to each element via CSS (more later).

To create a line break as opposed to a paragraph there is the `<br>` tag. Providing a closing pair for this tag is unnecessary. It can be either written as `<br>` or `<br/>`.

You **cannot** nest block element such as a `<p>` or `<h1>..<h6>` inside one another ie this is **WRONG**

```
<body>
<p>This is a <p>paragraph</p> of text.</p>
</body>
```

Semantics

Web designers often talk about 'semantic markup'. Semantics is the study of meaning. The heading tags are great examples of semantic markup as they are tags with meaning. That is a `<h1>` is heading one – the most important heading in the document. We'll see more semantic elements later.

Inline Elements

Inline HTML elements do not force a line break before or after them. They are most commonly used for basic formatting such as bold and italics.

Text can be made bold by using `<b>` or `<strong>`. The `<strong>` tag is considered to be more accessible for individuals using screen readers.

```
<p>
        <b>This text is bold</b>
</p>
<p>
        <strong>This text is bold</strong>
</p>
```

Text can be made italic by using `<i>` or `<em>`. The `<em>` emphasis tag is considered to be more accessible for individuals using screen readers.

```
<p>
     <i>This text is italic</i>
</p>
<p>
     <em>This text is italic </em>
</p>
```

Note: both bold and italics can be applied through CSS as we'll see later.

Adding Hyperlinks with <a>

The `<a>` anchor tag is responsible for adding hyperlinks. The `<a>` tag can be placed around both text and images to make them hyperlinks. The `<a>` tag needs attributes to make it do anything interesting. The main attribute associated with the `<a>` tag is `href` which stands for hypertext reference.

Notice there is a second HTML file called *qualifications.html*.

In *index.html* create a hyperlink to *qualications.html* using the following code:

```
<p>
<a href="qualifications.html">My Qualifications</a>
</p>
```

Understanding Absolute and Relative Paths

The above uses a 'relative' path. In web design we can link from one page to another using 'relative' or 'absolute' paths.

Relative Paths – A relative path finds its way to a particular file based on its starting point. Relative paths can therefore be shorter than absolute paths but care needs to be taken to ensure the route taken is correct. An example of a relative path is:

```
qualifications.html
```

If *qualifications.html* was in a sub-folder called *myStuff* then the relative would be:

```
myStuff/qualifications.html
```

If we linked from *myStuff/qualifcations.html* back to *index.html* then the path would be:

```
../qualifications.html
```

Absolute Paths – An absolute path is one that contains the full URL of the page to be linked to or the image to be included in the file. An absolute path leaves no room for dispute as to its location on the internet. An example of an absolute path is:

```
http://www.mywebsite/section2/introduction.html
```

Absolute paths are most commonly used when you link to a page external to your own web site.

When you make text a hyperlink it will appear blue and underlined. If you have visited that page already it will appear pink. The underline effect and colours

are default styling, that as we will see later, can be changed by the addition of some CSS.

Other attributes for <a>

The `<a>` will always need the `href` to be effective other attributes that can be used with the <a> are title and target.

With `title` the browser will reveal the text in the title when the user hovers of the link.

```
<a href="qualifications.html" title="My Qualifications">My Qualifications</a>
```

With `target` the browser will open the link in a new browser window when the value `_blank` is used. The option _new can also be used to limit the amount of tabs that are opened by targeted hyperlinks.

```
<a href="qualifications.html" target="_blank">My Qualifications</a>
```

HTML Lists

Lists are a very popular way of presenting information and with the addition of CSS can be styled to produce the likes of menu bars. The HTML for a list requires two elements `<ul>` (Unordered List) used to define the list itself and `<li>` (List Item) used to declare each item in the list. The `<ul>` is the parent of the `<li>`. That is the `<li>` tags need to nested inside the `<ul>` as follows:

```
<ul>
     <li>Apples</li>
     <li>Bananas</li>
     <li>Pears</li>
</ul>
```

The above would produce a list like this:

- Apples
- Bananas
- Pears

There is an alternative `<ol>` (Ordered List) tag that can be used to create ordered lists. However, we will see later that with CSS we can style up a `<ul>` list numerically without the need for an `<ol>`.

Both `<ul>` and `<li>` are block elements – in that their content begins on a new line. When we consider CSS you will see however that this default behavior can be changed.

Lists are often used to group links into navigation bars. As such it is common to see HTML such as:

```
<ul>
   <li><a href="index.html">Home</a></li>
   <li><a href="workExperience.html">Work
Experience</a></li>
   <li><a href="qualifications.html">Qualifications</a></li>
</ul>
```

Image File Formats

There are three main types of image available to include in a web site:

GIF files (Graphics Interchange Format) - An image format limited to 256 colours. Therefore GIFs are most useful for images with few colours or large areas of flat colours such as company logos, maps or charts. GIF files do support transparency but only one colour can be transparent. Becoming less popular with the rise of the PNG (see below). However they can be animated.

JPEG (Joint Photographic Experts Group) - You may be familiar with this format through digital photography as most digital cameras will save to this file format. JPEG files can be saved using a huge amount of different colours therefore making them ideal for photographs. JPEG do not support transparency.

PNG (Portable Network Graphic) - Designed to replace the GIF. Supports more than 256 colours and also multi-colour transparency.

Adding Images with <img>

Images are added with the `<img>` tag. The `src` attribute is used to indicate which particular image we want to use. The `src` is short for source and is the path to the image we wish to use. The path to the image can be absolute or relative.

Add the following to the index.html page.

```
<p>
     <img src="images/dishOne.jpg">
</p>
```

Other attributes that you are likely to use with an image include `width`, `height` and `alt`. The width and height indicate the dimensions of the image to be referenced and help speed up page loading. The `alt` attribute is used to provide alternate text for users who cannot see the image. It is also useful for improving your page's search engine ranking.

Attributes can be placed in any order but must be separated from each other by a single space. Therefore, a fuller example of the HTML required to insert an image would be:

```
<img src="images/dishOne.jpg" width="900" height="200"
alt="Jodrell Bank">
```

The values for width and height are in pixels. You do not need to explicitly declare them as been in pixels. As stated above the actual order of attributes does not matter but generally web designers will place 'src' first as this is the most important attribute to use with the `<img>` tag.

HTML Tables

HTML tables are defined using three core tags.

The `<table>` tag defines the beginning of the table and `</table>` the end. Each row is created with a `<tr>` element and ended with a `</tr>`. Cells within that row are created with the `<td>` (table data) element and ended with a closing `</td>`. It is inside the `<td>` tags that the actual content of the table is placed.

Therefore, to produce a simple three column, two row table the HTML would be as follows:

```
<table>
     <tr>
          <td>Column 1</td>
          <td>Column 2</td>
          <td>Column 3</td>
     </tr>
     <tr>
          <td>Value 1</td>
          <td>Value 2</td>
          <td>Value 3</td>
     </tr>
</table>
```

That would result in a HTML table as follows.

Column 1	Column 2	Column 3
Value 1	Value 2	Value 3

Table Headers

To make your table data more accessible when a `<td>` contains data that represents a column or row header then it is better to use the `<th>` element. The `<th>` table header tag is exactly like the `<td>` but is more semantic. It will also by default embolden and centre its content.

The Correct Use of Tables.

Tables have in the past been used by web designers as a way of laying out content. However, this technique dates back to the late 1990s when browsers did not have mature CSS support. CSS is now recognized as the best approach for web page layout. HTML tables are, however, ideally suited to displaying data. For example:

```
<table>
   <tr>
       <th>Qualification</th>
       <th>Subject</th>
       <th>Grade</th>
       <th>Date</th>
   </tr>
   <tr>
       <td>'A' Level</td>
       <td>Maths</td>
       <td>C</td>
       <td>August 2005</td>
   </tr>
   <tr>
        <td>'A' Level</td>
        <td>Physics</td>
        <td>A</td>
        <td>August 2005</td>
    </tr>
</table>
```

To make the table grid visible (and to stylize it) really requires CSS. We'll see more of this later.