HTML & CSS for Beginners

Section 1: HTML

1.1 Structure of a Website

When we enter a URL address of the website you are browsing in a web browser, a request is sent to the server where the website lives and if the request is successful it will send back a HTML code which is downloaded by your computer.

The web browser will read the HTML code, interpret it and render the webpage accordingly. We can use the browser's developer tool and select the Elements tab to view the HTML code of the website we are viewing. Below is an example for Wikipedia.

We can view this line by line to understand the HTML structure:

ELEMENTS/TAG	DESCRIPTION
<!doctype html>	Document Type tells the browser the type of document the webpage is. In this case html refers to HTML5. **Note:** There are other document types.
<html></html>	This is a HTML element tag. All tags have an opening and closing tag. The closing tag has a forward slash (/). We can add attributes to tags for example lang="mul" is a language attribute set to multiple as its attribute value.
<head></head>	The head element usually contains various pieces of information which are used by the web browser for different purposes. **Note:** If we expand the head element we can view other elements nested inside of this element.
<meta charset="utf-8"> <meta name="viewport" content="initial-scale=1.0, user-scalable=yes">	A meta data is some machine passable data which does not appear on the webpage but will add some extra information to the browser so that it can render the page in the correct way. **Note:** Some tags are self-enclosing and do not require a closing tag. **Note:** charset attribute sets the character encoding of the webpage e.g. utf-8 covers all the characters and symbols in the world. The viewport attribute is used by the browser to render the webpage to different device sizes. The user-scalable value allows the user to zoom in and out of the webpage.
<title></title>	This is a title element and the text appears in the Browser Page Tab. If you try to add the page to their favourites the title will be suggested to you.

ELEMENTS/TAG	DESCRIPTION
<link>	A link element is used to link the webpage to external resources. For example the <link rel="stylesheet" href="…"> links the webpage to an external CSS style sheet used for decoration purposes.
<body></body>	The body element contains everything we see on the webpage. **Note:** If we expand the body element we can view other elements nested inside of this element. When we hover over the elements we would see the corresponding element on the page being highlighted.
 <a> …	There are many different elements that can be nested inside of the <body> element. These tags make up the structure of what is displayed in the webpage for example: The tag is a divisional tag, <a> is a anchor/link tag, is a line break tag, <h1> is a header tag, bolds text, emphasises (italics) text, <small> makes text smaller, etc.

To add a comment to a HTML code you would use the following syntax (*adding comment where Comment Text is*):
!-- Comment Text -->

Comments are used by developers to make the code more readable and can be used by yourself or any other developer to help track the code.

This introduces the structure i.e. components which builds up the skeleton of a webpage.

The HTML code behind a webpage consists of text only. Therefore, to create a website all we need is a text editor to write some text. There are many text editors out there to choose from for example: VS Code, Atom, Sublime, Brackets, Notepad, Notepad++ to name a few. You can download and install these editors by navigating over to their webpage.

When creating a new file you would need to add the .html file extension so that the web browser can recognise the file is a website.

At the moment the file is sitting on your computer. If we were to open the file in the web browser you would notice that the URL address bar of the web browser is using the file protocol. This means that the browser is reading a file located on your computer.
We would want the file to be on a remote server so that it can be shared across the world. This is why we need web hosting. We can buy web hosting from various web hosting providers.

We can use a FTP (File Transfer Protocol) to send files across between your computer and your web server. A popular FTP application is FileZilla or Cyberduck. You will need your web hosting FTP credentials to setup a FTP between machines.

Once the file is transferred to the web host server, the webpage should be available to view for the whole world visiting the web host webpage URL assigned to you.

If we have a file called index.html inside the root public directory of the website, whenever a user enters the address of the website this will land the user in the content of the index.html file. If the file was named something else the user would need to enter the

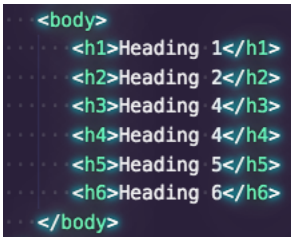
address followed by a forward slash (/) and the path to the file and the file name (for example: www.example.com/helloworld.html).

If there is no index.html file, when a user visits the web address (e.g. www.example.com) this will display the name(s) of the folders and files inside the public directory. Therefore, we would need to make sure we have one file called index.html inside of our web server for security reasons.

Note: There is another way to prevent the above from happening in the event of no index.html file inside the public directory. This will not be covered.

1.2 Headings, Paragraphs, Links and Images

The <h1><h1> tag is used to create a header element. The text between the header element tags is the header text that is displayed in the web browser.
There are 6 header levels each decreasing in size of the previous as the header level increases i.e. <h6> is the smallest header text size possible.



The <p> element tag is used to create a paragraph. The text between the element tags makes up the paragraph text and what ends up being displayed in the web browser. Every paragraph element starts on a new line. To create a line break inside of a paragraph we can use the
 line break element. For example:

```
<body>
    <h1>Heading 1</h1>
    <p>This is a paragraph.</p>
    <p>This is a second paragraph.</p>
    <p>This is a Third paragraph. <br /> With a line break to separate the text.</p>
</body>
```

The anchor tag element is used to create a link. The text between the element tag is the text that will be displayed in the web browser as a link.

The href attribute is used to tell the browser where the link should take us. The forward slash (/) at the beginning of the href attribute value refers to the root directory of the website. Each subsequent forward slash relates relates to sub-directory.

The title attribute value displays when you hover over a link element.

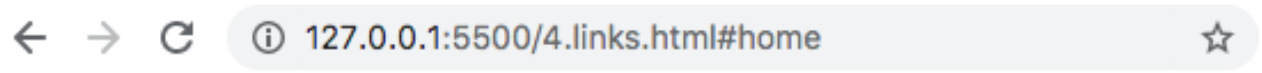
Link elements can be nested inside of other elements such as paragraphs and headings. You can link to another website or to another page in your own website.

```
    <p>This <a href="https://www.google.com" title="Google">L=link</a> will take you to Google.</p>
    <p>Return to <a href="/1.helloworld.html" title="Hello World">Home</a></p>
```

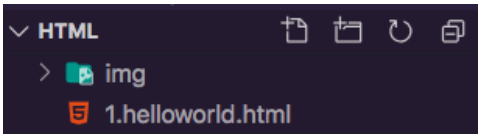
A link can be created to go to a subsection of the html document. This requires an id attribute on an element which the anchor element tag can reference. We use the pound sign (#) to reference a id attribute. This will only work where the element we wish to navigate to via the link is no longer visible at the top of the web browser view.

```
<h1 id="p1">Paragraph 1</h1>
<p>A very long paragraph text.</p>
<a href="#p1">Return to Top</a>
```

Note: When clicking on a sub-section link the web browser's address bar will display the id value at the end of the URL, example below #home:



We can create a sub-directory in our project file called images or img and store our image files within this folder.



To add an image to a html document we would use the element. This is a self-enclosing tag similar to the line break element.

The image element tag requires a few attributes. The first is the src attribute which tells the browser where it can find the source of the image. This can be a source to a directory path within the project directory or a link to a image hosted on a website. The image will be displayed in the full resolution size of the image.

The width and height attribute allows us to resize the image to a different size.
The alt attribute allows you to display an alternative text should the image not display for any reason e.g. the image no longer exists, file name/path has changed, etc.

```
<img src="/img/image.jpeg" alt="Feather" width="500px" height="500px">
<img src="https://www.princeton.edu/sites/default/files/styles/half_2x/public/images/2018/01/clouds-19.jpg?itok=Qfa5j6NW" alt="Clouds" width="1000px" height="500px">
```

These are some main body elements tags that can be used together to create the markup of the html document to display in the web browser.

1.3 Inline vs Block Elements

Elements can fit into two categories: inline and block.

Block elements takes as much height space as possible it needs while on the width it takes the whole line even if the content does not take up all the space on the width. The headings and paragraph elements are example of block elements.

Block elements start with a line break and end with a line break.

Inline elements on the other hand will only take as much space on both the height and width as it needs. The anchor/link and image elements are example of inline elements.

Inline elements that sit next to a preceding inline element will not have any line break. To create a line break between two inline elements a
 element can be used.

Inline element do not have a line break before them unless they come after a block element but the line break belongs to the block element.
Inline element do not have a line break after them unless they are followed by a block element but the line break belongs to the block element.

This is how we distinguish inline elements from block elements.

1.4 Iframes

Iframes are html documents which are embedded in other html documents. These elements are used to include any external content such as advertisements or YouTube videos etc.

Iframes are inline elements and do not have any line breaks between inline elements. This element requires a few attributes. The src attribute allows us to select a source for iframe. The source file will populate the iframe.

We can use the width and height element to resize the iframe. The frameborder attribute allows to set the border size.

```
<iframe src="3.paragraph.html" frameborder="1" width="500px" height="100px"></iframe>
```

We can add a link to a iframe which will allow a user to click on a link that will open the link inside of the iframe. To achieve this we require to use the name attribute on the iframe element.

Links have another attribute we can use called target. If target is set to _blank this will open the link in a new tab in your web browser. Alternatively, we can assign target an id name to target a html element.

```
<iframe name="myiframe" frameborder="1" width="500px" height="100px"></iframe>
<p><a href="https://www.wikipedia.org/" target="myiframe">Link</a></p>
```

Screenshot of before and after clicking the link element:

1.5 Unordered List, Ordered List and Description Lists

An unordered lists is a list of bullet points. To create a unordered list we would use the tag. In-between the unordered lists element tags we would use the element tag to populate the unordered list items.

We can nests <ul> elements within <li> elements to create a sub-unordered list.

```
<ul>
    <li>London
        <ul>
            <li>Cloudy</li>
            <li>Cosmopolitan</li>
        </ul>
    </li>
    <li>Madrid</li>
    <li>Paris</li>
</ul>
```

- London
  - Cloudy
  - Cosmopolitan
- Madrid
- Paris

An ordered list is a list of items which are ordered using letters or numbers. The syntax is the same as unordered list but we would use the tag for ordered list.

By default the list is ordered by numbers. We can use the type attribute in the ordered list element tag to change the ordered list type from number to letters. There are a few types i.e. 1 = number (default), A = uppercase alphabet, a = lowercase alphabet, I = roman numerals uppercase and i = roman numerals lowercase.

Again we can nest the unordered list as we did with unordered list.

```
<ol>
    <li>London
        <ol type="i">
            <li>Cloudy</li>
            <li>Cosmopolitan</li>
        </ol>
    </li>
    <li>Madrid</li>
    <li>Paris</li>
</ol>
```

1. London
   i. Cloudy
   ii. Cosmopolitan
2. Madrid
3. Paris

A description lists is a list of terms with a corresponding description pr definitions. To create a description list we need to use the <dt></dt> element tag to create the term and the <dd></dd> element tag to create the description.

```
<dt>HTML</dt><dd>- Stands for HyperText Markup Language</dd>
<dt>CSS</dt><dd>- Stands for Cascading Style Sheets</dd>
```

HTML
- Stands for HyperText Markup Language

CSS
- Stands for Cascading Style Sheets

These are the three distinct list types available to use within HTML documents.

1.6 Tables

The tables in HTML are similar to the tables that we can create using Microsoft Word. To create a table we would use the <table></table> element tags. Inside the tags are the content of the table.

The <tr></tr> element tags are nested inside of the table element to create the table rows. The first table row is generally used for the table headings i.e. column headers. We would nest the <th></th> element inside of the table row element.
Subsequent table rows are used for the table data. To create table data we would nest the <td></td> element tags in the table row element.

To the right is an example for creating a table in HTML and the following output displayed in a web browser.

CSS is used to style tables.

```
<table>
    <tr>
        <th>Name</th>
        <th>Company</th>
        <th>Date of Birth</th>
    </tr>
    <tr>
        <td>Bill Gates</td>
        <td>Microsoft</td>
        <td>1955</td>
    </tr>
    <tr>
        <td>Steve Jobs</td>
        <td>Apple</td>
        <td>1955</td>
    </tr>
</table>
```

Name	Company	Date of Birth
Bill Gates	Microsoft	1955
Steve Jobs	Apple	1955

1.7 Entities

HTML comes with many reserved characters for example the greater than (>), less than (<) and forward slash (/) signs. Another example is if we write some text inside of a <p> element tag and add some white text (space) this empty spaces will not display.

```
<p>This     is some text.</p>
```

This is some text.

To overcome the reserved characters we would need to use HTML entities. Entities are special characters we can write in our code to make the reserved characters appear on the webpage.

Entities are written using a & sign followed by either some text or a pound sign (#) followed by numbers and closed off with a semi-colon (;).

```
<p>This       is some text.</p>
<p>This       is some text.</p>
<p>This       is some text.</p>
```

This is some text.
This is some text.
This is some text.

The   entity is used to create a non-breaking space. We can also create this using numbers or hexadecimal values. Repeating the entity will create multiple non-breaking spaces rendered to the web browser. This allows us to add as many white spaces as we want in our text.

A reference page for all html entities can be found on https://dev.w3.org/html5/html-author/charref website.

1.8 Forms

Forms are very important to all applications because it allows the user to interact with the application. Forms are used for many purposes for example to log into your account, booking flights, searching for items and making payments on an e-commerce website, etc.

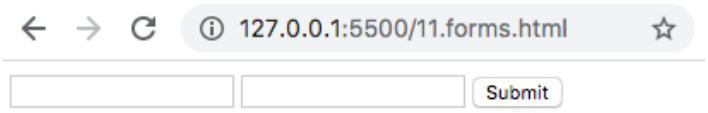
A form is a set of user inputs of fields which requires the user to populate. Once the form is submitted the fields will form a set of data which is sent to a server for processing.

To create a form the <form></form> element tag is used. All of the content of the form will sit inside of the form element tags.

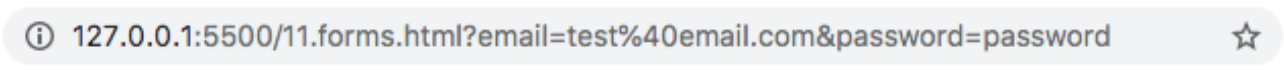
There are several ways of creating inputs depending on the type of input. We would use the <input> element tag which is a self enclosing tag and set the type attribute to whatever type of input we want for example text, password, submit, etc.
Input elements require a name attribute which will be used when submitting the form.
The input type of submit will create a submit button and this does not need a name attribute.

The form element requires a method attribute and there are many methods that can be used e.g. the get method. These are HTTP methods that we would use to send data across to the server.

```
<form method="get">
    <input type="text" name="email">
    <input type="password" name="password">
    <input type="submit">
</form>
```



When we submit a form we will notice in the address bar of the web browser there is a question mark (?) follows by all the input names and their value from the submitted form. The reason we would see this is because the form element tag has a GET method.



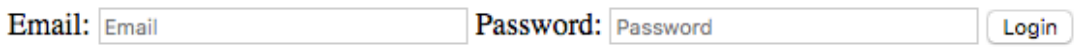
We can add more attributes to the input element such as the placeholder attribute which allows us to provide a placeholder text to provide hints for the user on completing the field.

The size attribute allows you to set a size fort he input while the maxlength attribute allows you to set a maximum length of characters allowed to be used in the input. The id element is used to allow other element tags to link to the input element by its id.

To change the default Submit text in the submit input element you can use the value attribute and changed the displayed text from the default.

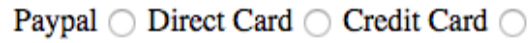
The <label for=“”></label> element tags can be used to add labels to the input fields. This element tag is placed before the input element. The for attribute is required to link the label to the input field. The value will be the input elements id value.

```
<form method="get">
    <label for="email">Email:</label>
    <input type="text" id="email" name="email" placeholder="Email" size="30px" maxlength="100">
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" placeholder="Password" size="30px" maxlength="30">
    <input type="submit" value="Login">
</form>
```



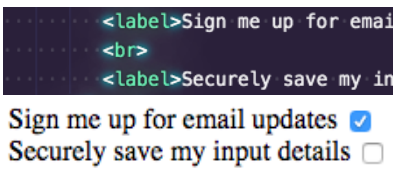
The input type of radio creates a radio button. This input type requires a name attribute and a value attribute.
The name attribute value are the same for the group of radio buttons. The value attribute for the selected radio button is passed to the server when the form is submitted.
Radio input elements must be nested inside of a <label></label> element in order to display the label text for the radio input.

```
<label>Paypal <input type="radio" name="payment" value="paypal"></label>
<label>Direct Card <input type="radio" name="payment" value="debit card"></label>
<label>Credit Card <input type="radio" name="payment" value="credit card"></label>
```



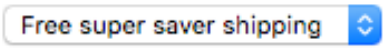
The checkbox input type is similar to the radio input type but the former allows the user to select more than one option whereas the latter only allows to select one option. Again we must wrap this input inside of a <label></label> element tag in order to display the text for the checkbox input type.
By default the checkbox is unchecked but you can change this by using the checked attribute setting its value to checked.

```
<label>Sign me up for email updates <input type="checkbox" name="email update" checked="checked"></label>
<br>
<label>Securely save my input details <input type="checkbox" name="save payment"></label>
```



The <select></select> element tag is used to create a dropdown menu. Inside the select tags we would nest <option></option> element tags to create a set of options inside of the dropdown input element. The option element requires a value attribute to pass on the data to the server so that the server code knows which option was selected. This can be named anything we want.

```
<select name="shipping">
    <option value="Free super saver shipping">Free super saver shipping</option>
    <option value="Standard 3 day shipping">Standard 3 day shipping</option>
    <option value="Next working day shipping">Next working day shipping</option>
</select>
```



The <textarea></textarea> element tag creates a text area for the user to enter free text. There are a few attributes that can be added to this element such as the name, id, cols and rows attributes. The cols and rows attribute are used to size the text area box.

```
<label for="comments">Comments:</label>
<textarea name="comments" id="comments" cols="40" rows="4"></textarea>
```

1.9 Text Decorations

Decorations and styling will be covered in Section 2: CSS; however, we can use simple HTML tags to surround text to add some decorations to the rendered text.

The element tag emphasises the text wrapped in the tag. The decoration will look different from one browser to another — the below is the decoration style within Google Chrome web browser.
The element tag decorates the wrapped text to display bold text.
The element tag decorates the wrapped text to display italic text.
The <strike></strike> element tag decorates the wrapped to display struck through text.
The <u></u> element tag decorates the wrapped to display underlined text.
The element tag decorates the wrapped text to display sub text.
The element tag decorates the wrapped text to display sup text.

```
<p>This <em>word</em> is emphasised.</p>
<p>This <strong>word</strong> is emphasised more. It stands out against text surrounding it.</p>
<p>This <i>word</i> is italic.</p>
<p>This <strike>word</strike> is struck through.</p>
<p>This <u>word</u> is underlined.</p>
<p>CO<sub>2</sub>...X<sup>2</sup></p>
```

This *word* is emphasised.

This **word** is emphasised more. It stands out against text surrounding it.

This *word* is italic.

This ~~word~~ is struck through.

This word is underlined.

CO_2...X^2

1.9 Comments

Comments are very important because it allows you and other developers to read code in an easy way. This is especially important for when you write code and come back to read it after a while.

To create a comment tag we would use <!-- Comment Text Goes Here -->
The comment/text goes in-between the double dash (-) signs. The comment text does not get rendered and displayed on the page. Comments can only be displayed inside of the text file within the text editor.

If using VS Code on a Mac, the keyboard shortcut for commenting is to hold down the cmd button and press the forward slash key (/) on your keyboard.
On Windows the shortcut is to hold down the ctrl button and press the forward slash key (/) on your keyboard.
Repeating the keyboard shortcut will toggle comment and uncomment the selected highlighted code.

```
    This is not a commented text
    <!-- This is a commented text -->
```

END