TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

INFO101 2020

Workshop 1

Introduction to HTML

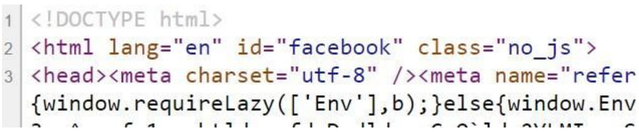
Contents

Introduction

Welcome to your first INFO101 Workshop! These guides and your workshops are designed to assist you as you learn various techniques for your INFO101 assignments and quizzes.

HTML

HTML (Hyper Text Markup Language), describes the structure of a webpage using markup language. HTML is very common on the internet, and is used by 73.7% of websites on the internet whose markup language is known. For example, if you view the source code of Facebook.com, you can see the source code of Facebook is built on HTML:



```
<!DOCTYPE html>
<html lang="en" id="facebook" class="no_js">
<head><meta charset="utf-8" /><meta name="refer
{window.requireLazy(['Env'],b);}else{window.Env
```

Figure 1: Example of Facebook.com HTML Code.

Because of this, learning HTML means you will be well equipped for how many websites work behind the scenes. Combined with Javascript and CSS, HTML makes up the structure of webpages on the World Wide Web (WWW), with HTML being the language for building web pages.

Software to Code Documents

On your computers or the SIM Lab computers you can create an HTML document using any Text Editor such as Notepad or Textedit. For these workshops, you should use Notepad++ which provides tools to help you to code HTML and CSS. These are installed on the SIM computers, but if you can download it for free on your device online.

We also advise that you use Chrome/Firefox as a web browser. Internet explorer often doesn't load everything & your work will be marked in Chrome/Firefox.

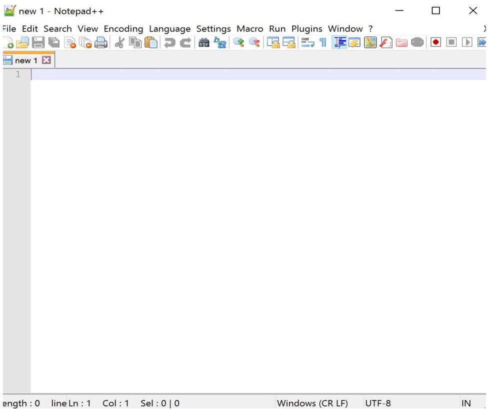
If you are using a Mac there are a number of options available. We recommend using a software called **Brackets** which is similar to Notepad++. Just go to **http://brackets.io/** or you can go to the Apple Store and download TextWrangler.

Getting Started

Opening Notepad++ (on Lab Computers)

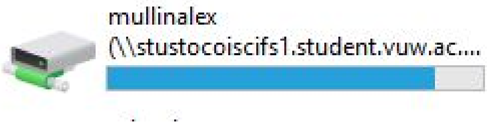
To open Notepad++ on the School of Information Management (SIM) Lab Computers:

1. Click Start on the Windows taskbar. The Start menu appears.
2. Start typing "Notepad" and click Notepad++ from the list that appears.
3. Open Notepad++ and you will see this:


new 1 - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1
ength : 0 line Ln : 1 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
IN

Saving an HTML File

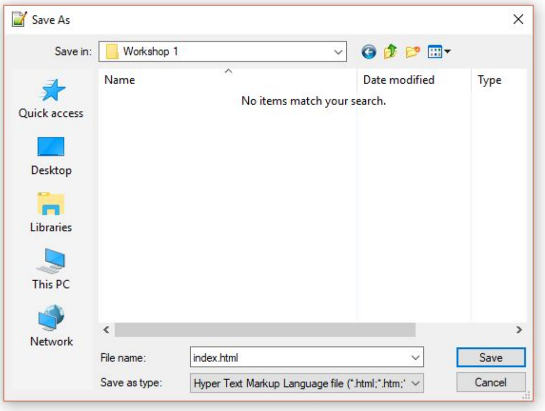
Next, you need to save a new HTML file. This should be saved inside of your **H Drive** (local drive with your name) on the university computers, or on a flash drive.



mullinalex
(\\stustocoiscifs1.student.vuw.ac....

Figure 2: Student H Drive

You can choose what to name this file, but we recommended to call it **'index.html'**. Also make sure you are saving the file type as a "Hyper Text Markup Language" type or .html for short.



Save in: Workshop 1

No items match your search.

File name: index.html

Save as type: Hyper Text Markup Language file (*.html;*.htm;*

Figure 3: Saving your html document

Re-opening your HTML File

If you exit out, to reopen your file, right click on the file and click 'Edit with Notepad++'

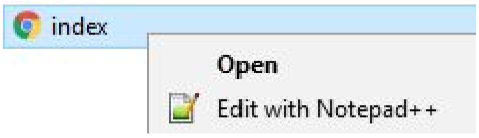


Figure 4: Opening with Notepad++

The Index Page

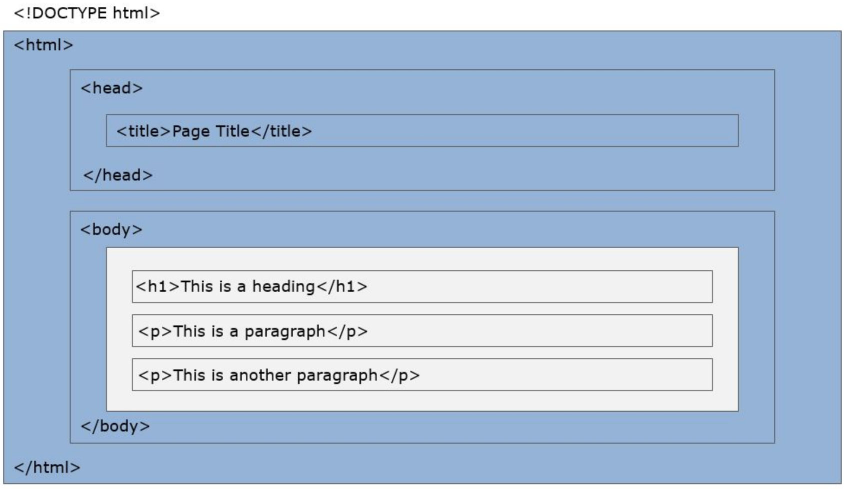
After doing the previous steps you may have noticed that we called the file **'index.html'**. You may be wondering why we called it this. In web development a fundamental rule in establishing the home or root page for a website we call it **'index.html'**. This has become a default setting within most directories and a standard expectation in coding.

Because of this, it will be expected that when you have a homepage, it will always be called **'index.html'** - **this is a requirement in the final assessment so keep this in mind.**

Coding Basics

The HTML Structure

In this workshop you will learn about the HTML structure and its necessary components. The diagram below outlines the basic structure in HTML and the sort coding elements and structure you can expect to learn.



```
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>
        <h1>This is a heading</h1>
        <p>This is a paragraph</p>
        <p>This is another paragraph</p>
    </body>
</html>
```

Figure 5: DOCTYPE and HTML tags

The white content is what is visible in the web browser

DOCTYPE and HTML Tags

At the top of every HTML page there is a basic line that enables you to format your document correctly. There are two main parts: **DOCTYPE** and **HTML** Tags.

At the beginning of every HTML document you need to tell the web browser what type of document it is reading. What is required is an HTML declaration of the code type. Use the following declaration on the first line:

<!DOCTYPE html>

After the DOCTYPE tag and at the end of every HTML page are the opening and closing **HTML tags. HTML requires a forward slash / in its closing tags**.

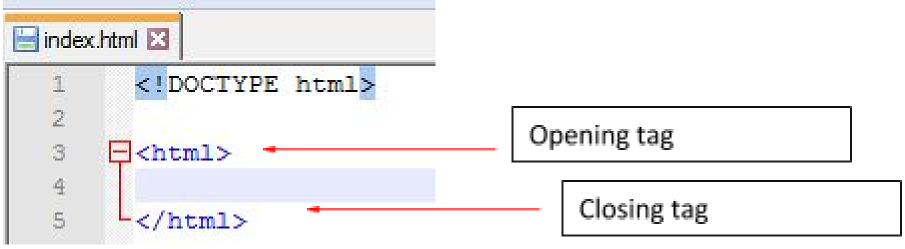
All of your HTML code needs to be between the opening and closing <html> tags.

<html>

(Your Website content)

</html>

Add these HTML tags to your file, below the HTML declaration '**<!DOCTYPE html>**'. You do not need to include the content between of the brackets (yet), this is just for example.



```
<!DOCTYPE html>

<html>

</html>
```

Opening tag

Closing tag

Figure 6: DOCTYPE and HTML tags

HTML Comments

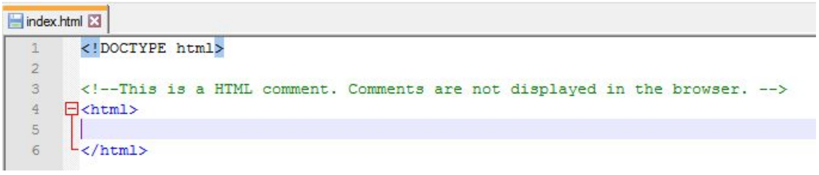
An important habit to pick up is including **comments** inside of your HTML. Comments are written in your code, but are not displayed by the web browser. They are useful for explaining why you have coded something.. You can place comments anywhere in your code, and our easily distinguishable by green text.

Try an HTML comment for yourself. Above your opening HTML tag, add the following comment:

```
<!-- This is a HTML comment. Comments are not displayed in the browser. -->
```

Figure 7: An example of an HTML comment.

Which should fit into your HTML document directly above what you are commenting on. In this instance, right above the opening HTML tag. Your HTML document should now look like Figure 7:



```
<!DOCTYPE html>

<!--This is a HTML comment. Comments are not displayed in the browser. -->
<html>

</html>
```

Figure 8: Comment above the opening HTML tag (normally you would change to suit content)

Note: You can also use comments to "block" any code that you don't want the application to read.

Head and Body Tags

Tags are very important. All HTML pages are broken up into two main sections – the head and body section. Each of these sections are defined by the associated tags: **<head> </head>** and **<body> </body>**.

The **head** section is not to be confused with a **heading** as these are separate elements. The **head** contains important metadata and about the document such as the page 'title' (see figure 10). It is also where you place the code to link your CSS file to your HTML (we will learn this in Workshop 2). Like all tags, at the end of your head section, place a closing </head> tag.

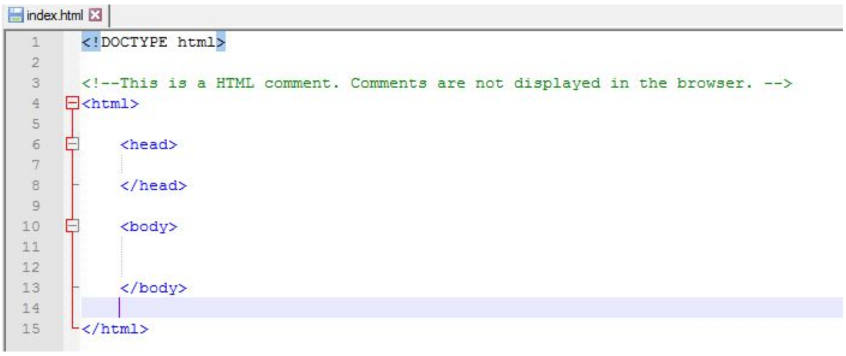
Place the opening head tag <head> after the **opening** <html> tag. This will be nested between your <html> tags. (see figure 9)

The **body** section is where your pages content goes. The <body> tag designates the beginning of the actual content. **Everything that is displayed and can be viewed on the web page will be enclosed in**

the <body> tags. This is important to know, if there is content outside of the <body> tags in your assignment, you will lose marks for structure.

Place the <body> tag after the **closing** </head> tag.

There should be no content in between the two sections! All code should be within tags. At the end of your document place a closing </body> tag directly **before the closing** </html> tag.



```
<!DOCTYPE html>

<!--This is a HTML comment. Comments are not displayed in the browser. -->
<html>

    <head>

    </head>

    <body>


    </body>

</html>
```

Figure 9: Nested HTML tags at this point in the workshop.

Note: To "nest" tags press "Tab" at the start of the intended line.

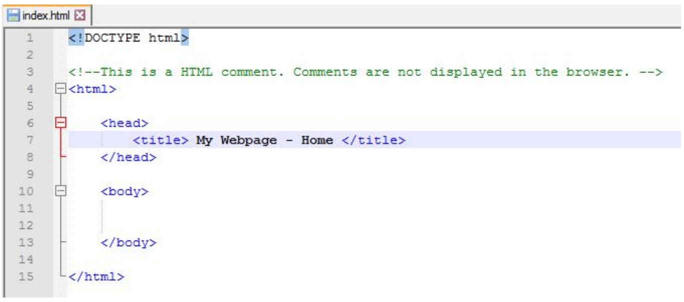
Giving your Webpage a Title

The next step is to now add a title. Every HTML page should have a title. The title appears in the browser's **Title bar**, which appears at the top of the browser's window or tab bar. **The title is not a heading as this doesn't display on your web page**.

Titles are created using Title tags <title></title>. These **Title** tags are placed within the **head section** of your document. **Here are some tips when writing titles:**

- Letters and numbers can be included in your title.
- A title cannot contain formatted text or images.
- Use a common element to start related pages: e.g. XYZ Company – Annual Report
- Your document can have only one title tag

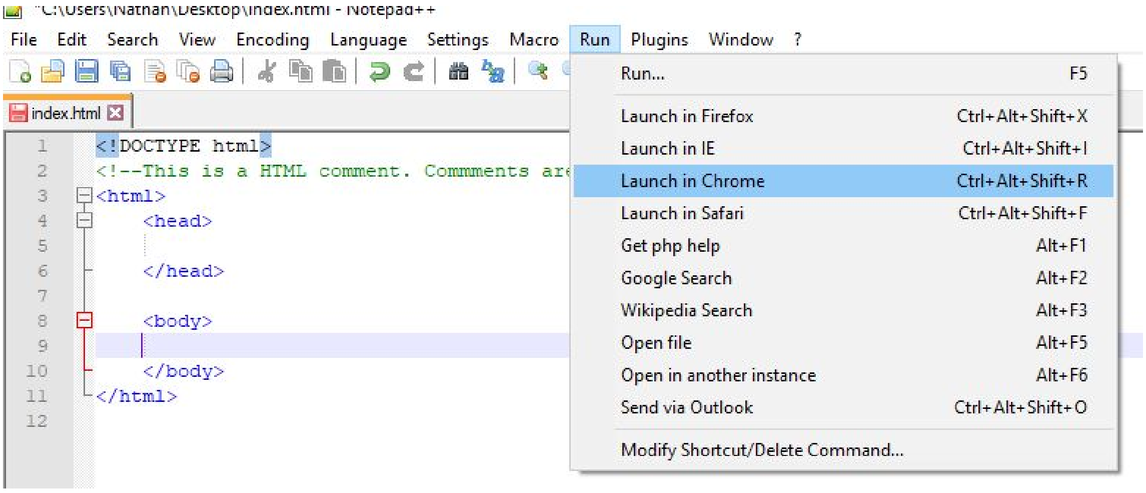
To create your title, make sure you have opened your **index.html** in **Notepad++.** On the line below your opening <head> tag, type: **<title> My Webpage - Home </title>**. Your document should look like the picture below:



```
<!DOCTYPE html>

<!--This is a HTML comment. Comments are not displayed in the browser. -->
<html>

    <head>
        <title> My Webpage - Home </title>
    </head>

    <body>


    </body>

</html>
```

Figure 10: Example of Title tag.

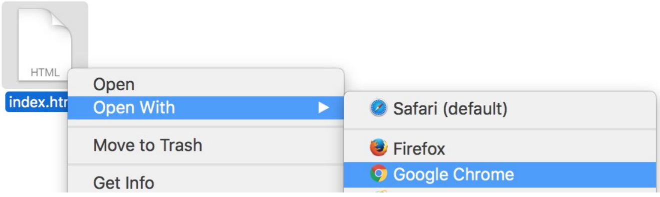
To test your HTML file, first make sure you save your file, and then either click 'Run' and 'Launch in…' Google Chrome;


*C:\Users\Nathan\Desktop\index.html - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
index.html
Run... F5
Launch in Firefox Ctrl+Alt+Shift+X
Launch in IE Ctrl+Alt+Shift+I
Launch in Chrome Ctrl+Alt+Shift+R
Launch in Safari Ctrl+Alt+Shift+F
Get php help Alt+F1
Google Search Alt+F2
Wikipedia Search Alt+F3
Open file Alt+F5
Open in another instance Alt+F6
Send via Outlook Ctrl+Alt+Shift+O
Modify Shortcut/Delete Command...

```
<!DOCTYPE html>
<!--This is a HTML comment. Commments are
<html>
    <head>

    </head>

    <body>

    </body>
</html>

```

Figure 11: Example of Title tag.

Mac Users: ***If you are using TextWrangler there isn't an option to 'Run' your website. To test on mac, go into the folder that it is saved in, right click on the file and open it in Google Chrome/Safari:***



When your website is open the file will have the title in the tab:

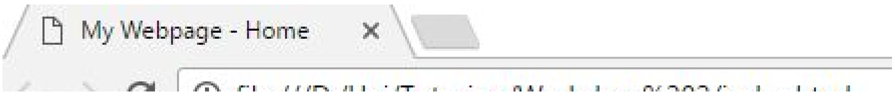


Figure 12: Title as it appears in a web browser

TRY: Try changing the words within your title tags, then 'Run' your webpage again, to see what happens!

Coding your content

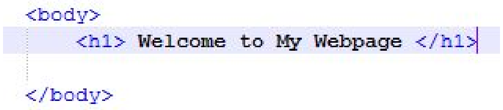
In order to bring content to your webpage you will be working within the **<body> </body> tags**.

Adding Text to your Webpage

The most basic element of the body section of a webpage is to enter text and code that will been shown on your webpage. **This content should all be within your body tags <body></body>.**

Heading Tags

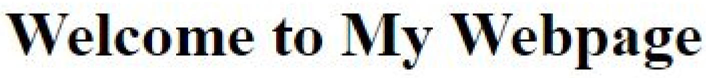
First, inside of the body tags, enter the heading for your webpage, using the **<h1> </h1>** tags. Wrap the text ‘Welcome to My Webpage’ in the heading tags:



```
<body>
    <h1> Welcome to My Webpage </h1>

</body>
```

Figure 13: Code to create headings tags.

When you run your web page it should look like this:

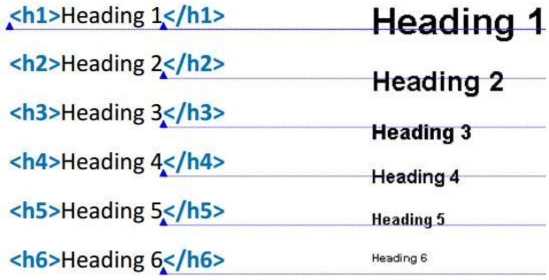


Welcome to My Webpage

Figure 14: Testing the <h1> tag.

You can adjust the size by changing the *h1* to any other number, up to 6, which will decrease the size. ***Note: the heading tag cannot exceed 6.***

<h6> is the smallest heading, <h5> is the second smallest, etc. This is demonstrated in Figure 11:



```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Figure 15: The 6 different types of <h> tags, and their respective sizes.

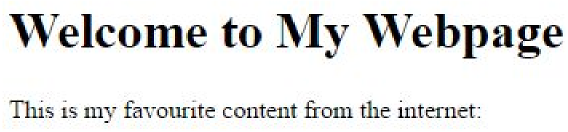
TRY: Try using different heading tags, then ”Run” the page to see the differences.

Paragraph Tags

We use <p> tags when adding regular content in paragraphs. In the body add some text e.g. ‘This is my favourite content from the internet:’. Make sure to put this text in paragraph <p> tags. Place a <p> tag at the beginning, and a </p> at the end.

Note: If you are using more than one paragraph you can use the <p> tag as many times as you like. ***<p1> is not a tag.***

Once that is done, run your website. You should now have something like this:



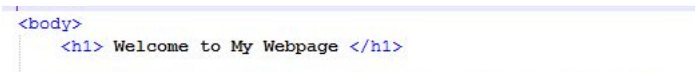
Welcome to My Webpage

This is my favourite content from the internet:

Figure 16: Example of <h1> tag and <p> tag.

Indenting:

You may have noticed that in the examples the tags in the code aren't aligned the same. The reason we do this in code is to keep it tidy and easier to interpret a hierarchy in the tags. By doing this it makes the code easier to read, especially when you have a large amount of code. You can do this by clicking the "tab" key.



```
<body>
    <h1> Welcome to My Webpage </h1>
```

Figure 17: Indented text

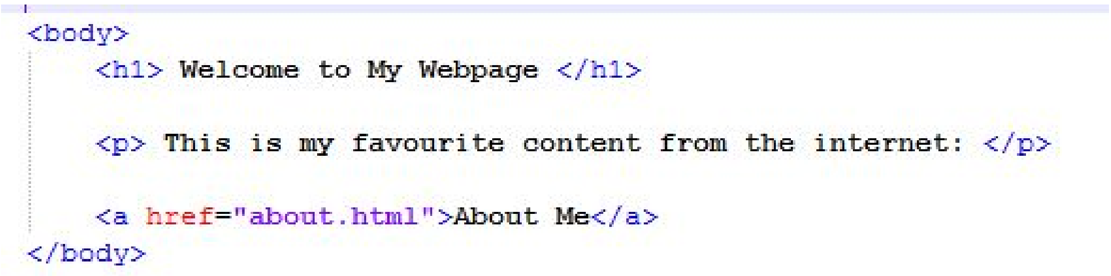
Creating Links

Creating a Hyperlink to a Local HTML Page (Internal Link)

With your homepage created on index.html, you can create new HTML pages to link to your original page. First, you will need to create a new HTML page using the same method as you did for the first one, but instead of index, call is 'about'

The '**about' page will be a separate** HTML file. Save the new HTML file as 'about.html' **in the same file location as index.html**. If this file is not saved in the same location the link won't work. Remember to save the new .html file as the correct file type, as shown earlier.

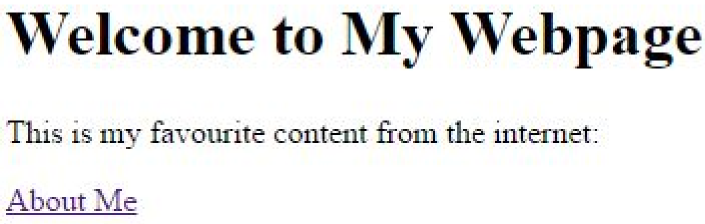
To link an external source inside of an HTML file the ***href*** attribute is used. This attribute is used to specify the destination of a hyperlink address. Inside of your body in index.html, enter the following HTML code to create a hyperlink, **make sure that you put the exact file name of your new page**:



```
<body>
    <h1> Welcome to My Webpage </h1>

    <p> This is my favourite content from the internet: </p>

    <a href="about.html">About Me</a>
</body>
```

Figure 18: Inserting a local link using <a href="about.html">

The text 'About me' is what people will see, and click on, to access this page. Test your index.html page. You should see the 'About me' text:



Welcome to My Webpage

This is my favourite content from the internet:

About Me

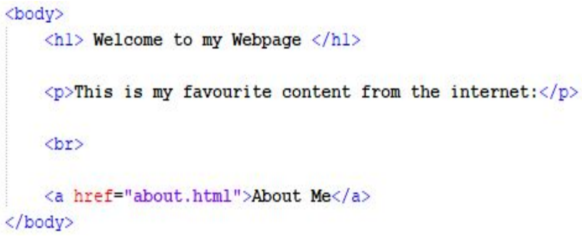
Figure 19: An *href* attribute from the browser view.

The text is underlined and a different colour. This indicates that the text is a hyperlink which can be clicked on. When clicked, the browser should direct you to the *about.html* page, which will be blank at this point as we have not put anything there yet.

Line Breaks

If you want to move text on the next line down or add another line space, use the Line Break tag.

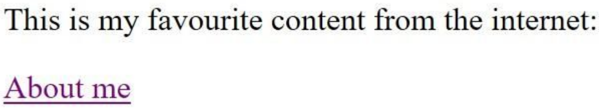
At the end of your <p></p> add the tag **<br>.**



```
<body>
    <h1> Welcome to my Webpage </h1>

    <p>This is my favourite content from the internet:</p>

    <br>

    <a href="about.html">About Me</a>
</body>
```

Figure 20: Inserting the <br> tag

If you run your website your hyperlink should shift down.



This is my favourite content from the internet:

About me

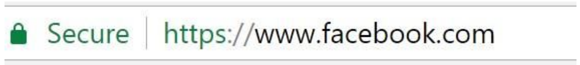
Figure 21: Text using paragraph tag.

The differences may be subtle so add more <br> tags to see stronger results. This is useful for spacing content/images.

Creating a Hyperlink from a URL (External Link)

The URL is just a fancy name for address of a website. It contains information about where to find a file, and what the browser should do with it once it has found it.

Every page on the Internet must have a unique URL. Without it, the document cannot be found on the network. For example, to access Facebook, the URL is required:



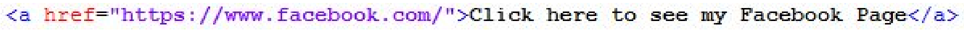
Secure | https://www.facebook.com

Figure 22: The URL to access Facebook.com on the Internet.

When you create a link to a webpage you need to know the address or URL of the webpage. To create a link, use the href tag from earlier:

<a href="address">Clickable text</a>

Position this below your 'About me' text. Change the text 'address' to be the URL you want to link. For example, if you wanted to re-direct someone to Facebook, you could use this clickable text:



```
<a href="https://www.facebook.com/">Click here to see my Facebook Page</a>
```

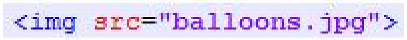
Figure 23: The href code to access Facebook.com on the Internet.

Be sure to substitute the "address" in with the appropriate URL address. To check if it works, run your index.html and if you click the link it should take you to Facebook.

TRY: Try linking to other webpages and changing the clickable text to fit with the new link location.

Images

Adding images from your computer is fairly easy. First, search for an image that you like online and save it in **the same folder as your HTML file**. Right click on an image, and click 'Save image as…'. Remember to **give it a name that is easy to remember about the image** e.g. we used a picture of balloons so called it balloons.jpg

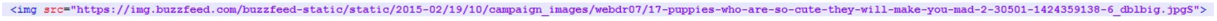


```
<img src="balloons.jpg">
```

Figure 24: The image tag using an image saved in your folder.

The image tag is special because this is one of the only tags that you don't have to close.

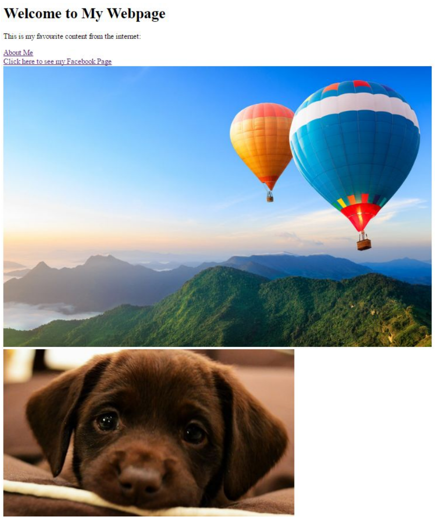
In this example we used something saved on the computer, however this part can be substituted by a URL of an image. The src (or source) attribute simply designates the location and name of the image. The file location could be on the internet. To link an image from the internet, find another image online, right click on it and select 'copy image address'. Rewrite the image tag below your first one and after the 'src' paste the URL.



```
<img src="https://img.buzzfeed.com/buzzfeed-static/static/2015-02/19/10/campaign_images/webdr07/17-puppies-who-are-so-cute-they-will-make-you-mad-2-30501-1424359138-6_dblbig.jpgS">
```

Figure 25: The image tag using the URL of an online image.

If you test it, your web page should look something like figure 26.



Welcome to My Webpage

This is my favourite content from the internet:

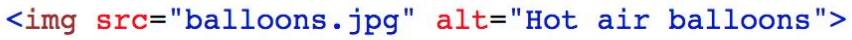
About Me
Click here to see my Facebook Page

Figure 26: Webpage with images.

Remember to use <br> tag to line break if you do not want your content to be side by side.

‘Alt’ Text

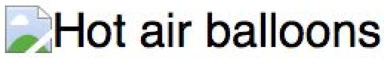
An additional attribute to add to the <img> tag is alternate text. Some browsers do not support certain image types, or images may have corrupted and fail to load. When this happens, ***alternate image text*** is useful to tell the user what the image actually is. For example, the alternate text has been added to the above HTML picture:



```
<img src="balloons.jpg" alt="Hot air balloons">
```

Figure 27: Image tag with alternate image text.

If a user cannot view the image in the browser, they will see this instead of the image:



Hot air balloons

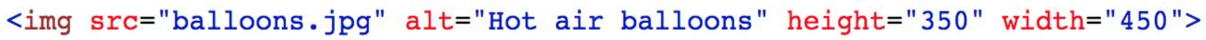
Figure 28: The alternate image text, substituting in for an image.

NOTE: For your assignments you will be required to use alt text for you images, however it needs to be relevant to the image used e.g. if the ALT text is for a balloon it could say “balloon”.

Image Sizing

Now that you have your image(s) we will work on sizing them.

Adjusting height and width of images can be useful. Inside of your img tag you can tell HTML the height and width you would like your picture to be. Add the following to your image tag, specifying the height and width of the image with the attributes “height” and “width” e.g. a height of 350 and a width of 450:



```
<img src="balloons.jpg" alt="Hot air balloons" height="350" width="450">
```

Figure 29: Image tag including height and width

You can do this for the other image as well. By keeping the size similar, it makes them uniform in size.

Embedding content

Videos

To include a video frame find a video you want to embed, for example on Youtube. Once you have found a video click on the share button (located above or below the subscription bar for the channel:

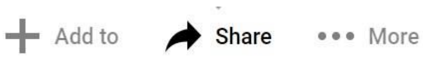
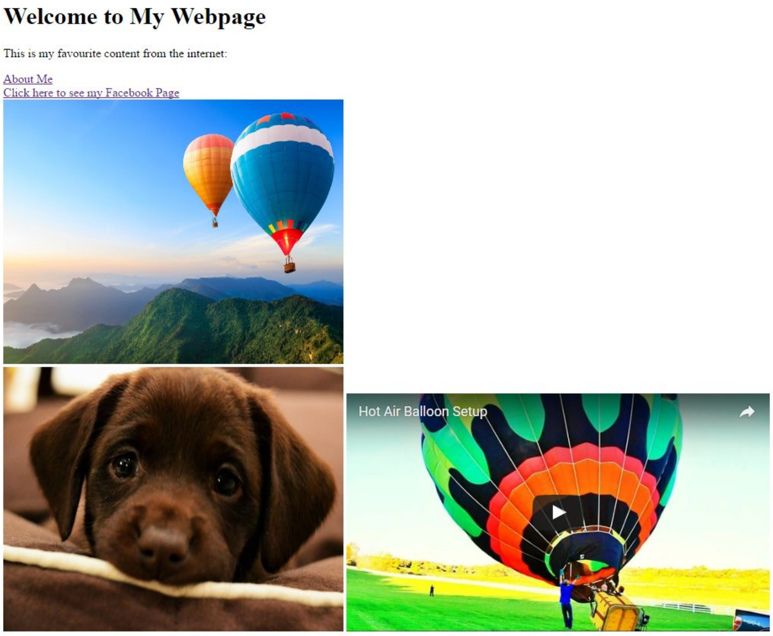


Figure 30: Share button on Youtube.

Once you have clicked onto the share button you will have three options: 'Share', 'Embed' or 'Email'. Choose 'Embed'. You will be provided with HTML code in an **<iframe>** tag. **You do not need to edit this code**. Copy and paste this into your HTML body and the video should appear on your page:



Welcome to My Webpage

This is my favourite content from the internet:

About Me
Click here to see my Facebook Page

Figure 31: Example of the iframe HTML code in the browser.

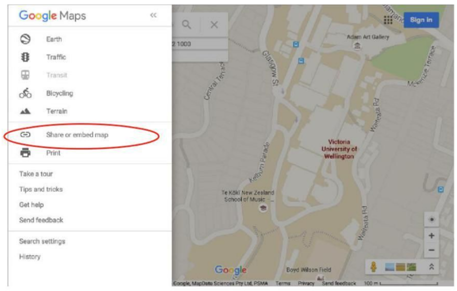
If you wanted to adjust the size of the video, you will notice within the iframe tag there are height and width values, you can change those accordingly; and remember to use <br> tag if necessary.

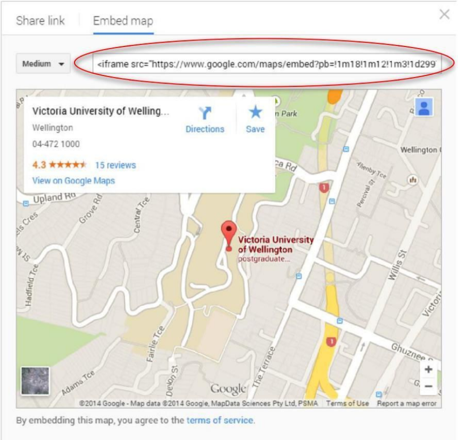
We encourage you to experiment with these codes, and even research other codes or other attributes you may find useful.

Embedding a Google Map

You can embed a basic map of a particular destination on Google Maps.

1. Search a location in Google maps
2. Ensure that the map you'd like to embed appears in the current map display.
2. In the top left corner, click the main menu ☰ .
3. Click Share or embed map.
4. In the box that appears, make sure the Embed map tab is selected.
5. Choose the size you want, then copy the code and paste into the body of your html document.





```
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d299
```

By embedding this map, you agree to the terms of service.

Depending on what internet browser you are using and version it is, the location and look of the settings icon may differ.

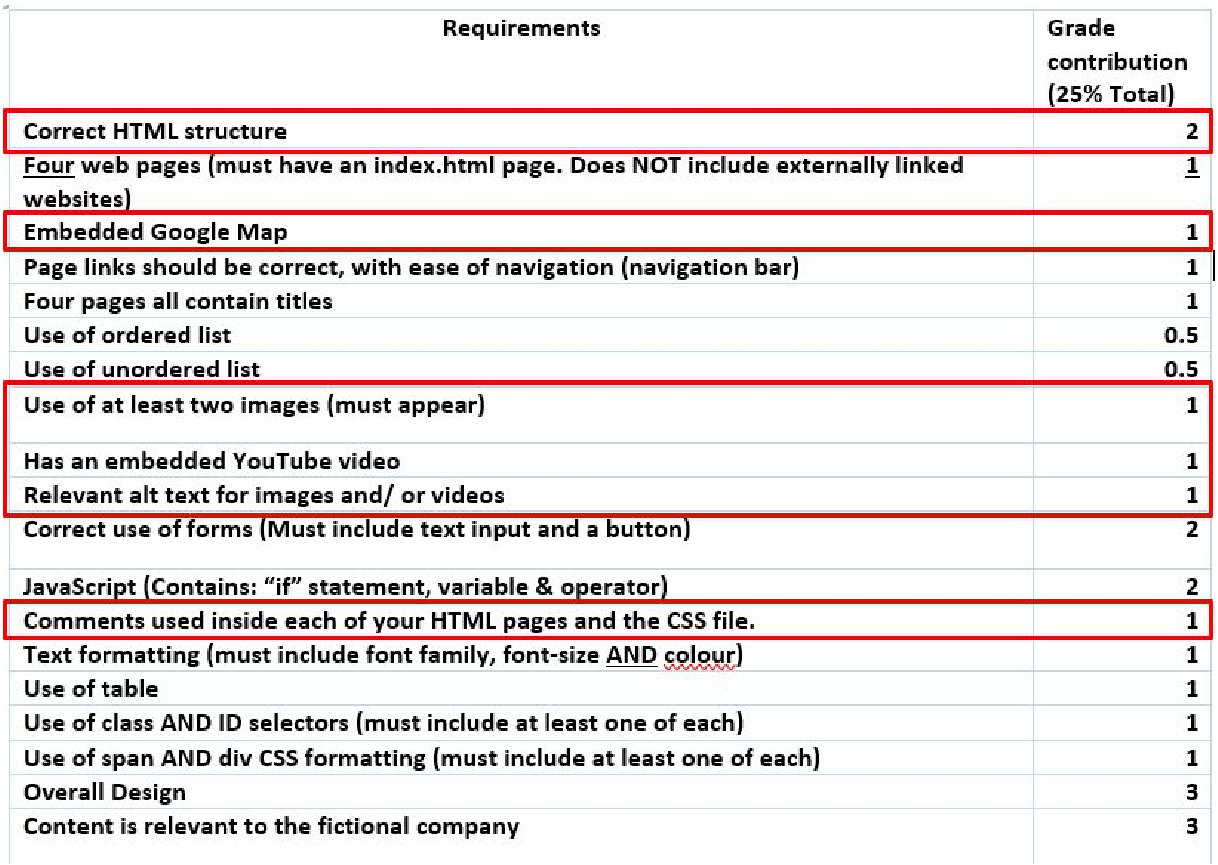
Putting things into practice:

Now that you have the skills to start a basic page why not try something from scratch?

Go onto Blackboard -> INFO101 -> Assignments -> Project -> Website Project.

Read the Website Project Brief and try and apply the skills you have learned. (**Note:** For students who are new to HTML, CSS and Javascript, this is not a project that can be easily completed over night, it is recommended you use the Workshop time to start your project and ask the tutor's any questions you may have.)

Below is the Website Project Rubric, now that you have finished Workshop Guide 1, you should be able to complete the highlighted sections below:



Requirements	Grade contribution (25% Total)
Correct HTML structure	2
Four web pages (must have an index.html page. Does NOT include externally linked websites)	1
Embedded Google Map	1
Page links should be correct, with ease of navigation (navigation bar)	1
Four pages all contain titles	1
Use of ordered list	0.5
Use of unordered list	0.5
Use of at least two images (must appear)	1
Has an embedded YouTube video	1
Relevant alt text for images and/ or videos	1
Correct use of forms (Must include text input and a button)	2
JavaScript (Contains: "if" statement, variable & operator)	2
Comments used inside each of your HTML pages and the CSS file.	1
Text formatting (must include font family, font-size AND colour)	1
Use of table	1
Use of class AND ID selectors (must include at least one of each)	1
Use of span AND div CSS formatting (must include at least one of each)	1
Overall Design	3
Content is relevant to the fictional company	3

TRY: Try playing around with the different tags and text. If you want to explore and try some other html tags then you can go to https://www.w3schools.com/html/default.asp.

Glossary of tags & terms:

Terms:	**Definition:**
HTML	Hyper text markup Language.
Tag	Names of elements surrounded by angle brackets.
Element	The basic building block of HTML and is typically made up of *two tags*: an opening tag and a closing tag.
Wrapping	When an element is opened and closed.
Nesting	When an element is contained inside another element.
Internal Link	hyperlink on a webpage to another page or resource, on the **same** website.
External Link	hyperlink on a webpage to another page or resource on a **different** website.
Embedding	Inserts external elements onto a webpage e.g. video or map.
Attribute	Attributes provide additional information about HTML elements e.g. href.

Tag:	**Purpose:**
<!DOCTYPE html>	Represents the document type, and helps browsers to display web pages correctly.
<html>	The root structural element of an HTML page.
<head>	Contains meta information about the document (no actual page content).
<body>	Structural element that **contains the visible page content.**
<title>	Establishes what will be seen in the browser's tab.
<h1>	Defines a large heading.
<p>	Defines a paragraph.
<br>	Line break.
<a>	Defines a hyperlink, which is used to link from one page to another.
<img>	The image element.
Attribute:	**Purpose**
alt	Specifies an alternative text for an image, when the image cannot be displayed.
href	Specifies the URL (web address) for a link.
src	Specifies the source and name of an image to be displayed.