Create a simple HTML Page

Overview

Using the basics of HTML markup, you will create a simple webpage using a code editor. Note that we are only using simple markup this week to create structured content. Don't worry the visual appearance of the page - CSS formatting will be covered in a future lab activity.

For this week's exercise make sure you include the following sites:

- A main title for the page
- Logically structured headings and sub-headings
- Correct use of paragraphs and line breaks
- At least 1 special character using an HTML entity
- Some text with emphasis and strong emphasis
- Lists (ordered or unordered)
- A table containing some data
- Images and hyperlinks using absolute and relative URLs

This exercise will use text content from the following sources:

- **Wikipedia:** https://en.m.wikipedia.org/wiki/Never_Gonna_Give_You_Up
- **AZ Lyrics:** http://www.azlyrics.com/lyrics/rickastley/nevergonnagiveyouup.html

You can also use your own original text content, or text from another source, if you prefer.

Initial setup

1. Create a folder on the desktop (or your own disk) named **week3**.

2. Inside the **week3** folder, create a new sub-folder named **images**.

3. To begin creating your first page, open a code editor application on your computer. In the faculty PC labs you will have access to:

    - Adobe Brackets
    - Microsoft Visual Studio Code

    You can also download these applications for to use on your personal devices from https://brackets.io/ and https://code.visualstudio.com/.

4. Using your code editor, create a new file and save it in your **week3** folder as **index.html**.

Setting up required HTML structures

1. In your editor, start by adding an **html** element consisting of a start and end tag:

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

    This tells web browsers that this entire document should be treated as an HTML page.

2. Between the **html** tags, add **head** and **body** elements, each with their own start and end tags:

```
<html>

    <head>
    </head>

    <body>
    </body>

</html>
```

The **head** element will hold the **metadata** of the document. This will provide information that web browsers can use that will not be shown as visible content in the window.

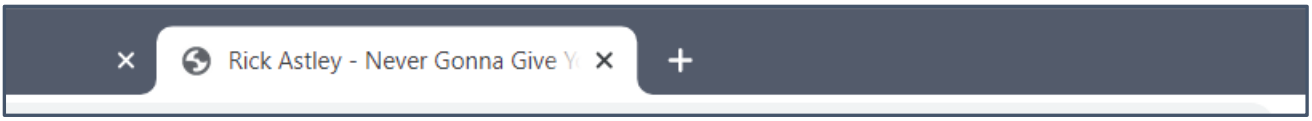
The **body** element will be used for all of the visible page content.

Note how both of these elements are indented? This allows you to easily see that these elements are **children** of the **html** element. Consistent indenting keeps your code readable. You can easily change the indent of multiple lines of code by selecting the lines and pressing **tab** on the keyboard to move the indent inwards. Press **shift+tab** to move the indent back out.

3. Add a **title** element within the **head** tags and page title text between the start and end tags:

```
<head>
    <title>Rick Astley – Never Gonna Give You Up</title>
</head>
```

Save your page, then go to your folder and open the page in Google Chrome. The empty page will be displayed with the title shown as a label in the tab:



Every page should have a descriptive title! The title is also displayed for browser bookmarks, history and search engine results

Add some basic text structures

1. Make some headings between **body** tags using heading elements:

```
<body>

    <h1>Rick Astley</h1>
    <h2>Never Gonna Give You Up</h2>
    <h3>Lyrics</h3>
    <h3>Cover Versions</h3>
    <h3>Rick-rolling</h3>

</body>
```

Make sure each heading element is closed with a matching end tag.

You may notice that **h3** was used multiple times – this means that these sub-headings are **equally third-most important**.

HTML allows up to 6 levels of headings (**h1** through to **h6**) to be used in a document.

2. Inside of the **body** element, copy a few short paragraphs of text from the Wikipedia article and paste them under the **h2** heading with blank lines in between each paragraph.

```
"Never Gonna Give You Up" is a song recorded by British singer and
songwriter Rick Astley, released as a single on 27 July 1987.

In 2004, "Never Gonna Give You Up" was voted number 28 in 50 Most Awesomely
Bad Songs... Ever by VH1.

The music video for the song has become the basis for the "Rickrolling"
Internet meme, leading the song to also be referred to as "The Rickroll
Song".
```

Save the page and reload it in your web browser (press **Ctrl+R** or **Cmd+R**). You will see the text displayed with no spacing between the paragraphs.

3. Turn the text into separate paragraphs, each piece of text needs to be surrounded in **p** elements:

```
<p>"Never Gonna Give You Up" is a song recorded by British singer and
songwriter Rick Astley, released as a single on 27 July 1987.</p>

<p>In 2004, "Never Gonna Give You Up" was voted number 28 in 50 Most
Awesomely Bad Songs... Ever by VH1.</p>

<p>The music video for the song has become the basis for the "Rickrolling"
Internet meme, leading the song to also be referred to as "The Rickroll
Song".</p>
```

4. You can also use **br** elements to manually insert a break into a line. Try using this to add a paragraph under the **Lyrics** sub-heading:

```
<p>Never gonna give you up<br>
Never gonna let you down<br>
Never gonna run around and desert you<br>
Never gonna make you cry<br>
Never gonna say goodbye<br>
Never gonna tell a lie and hurt you</p>
```

In your browser, you should see that each line of the lyrics is displayed a separate line, without spaces between them.

The **br** tag is an example of an **empty element**. It has no closing tag but can be **self-closed**.

```
<p>Never gonna give you up<br />
```

5. Some characters are difficult or impossible to type on a keyboard. Instead you can create these characters with special character codes called **HTML entities**:

Character	Entity Code
©	©
™	™
®	®
€	€

Character	Entity Code
♠	♠
♣	&clubs
♥	♥
♦	♦

Character	Entity Code
<	<
>	>
&	&
=	=

Some entities can be used to display characters that might conflict with the syntax of HTML itself.

View the full list of HTML entities: https://dev.w3.org/html5/html-author/charref

Try using an entity code to display copyright information underneath the lyrics:

```
<p>Lyrics &copy; 1987 Stock Aitken Waterman and RCA Records</p>
```

6. Some HTML elements can make **in-line** structures around content – that is, content that does not get forced onto a new line, but instead will sit in-line with other content.

   Use **strong** and **em** elements to make some of the important words in the earlier paragraphs stand out more:

```
<p><strong>"Never Gonna Give You Up"</strong> is a song recorded by British
singer and songwriter Rick Astley, released as a single on 27 July 1987.</p>

<p>In 2004, <strong>"Never Gonna Give You Up"</strong> was voted number 28
in <strong>50 Most Awesomely Bad Songs... <em>Ever</em> by VH1</strong>.</p>

<p>The music video for the song has become the basis for the
<em>"Rickrolling"</em> Internet meme, leading the song to also be referred to
as <em>"The Rickroll Song"</em>.</p>
```

   There are 2 elements being added to the paragraphs here:

   - **em** Indicates *emphasis*, with the default appearance of *italic text*.
   - **strong** Indicates **strong emphasis**, with the default appearance of **bold text**.

   In this example the word **"Ever"** is within an **em** element that is itself nested within a **strong** element. This creates ***very strong emphasis*** that is displayed as both ***italic and bold***.

Create a list of information

List elements are useful for displaying **itemised sets** of information. In HTML, lists can be **ordered** (a numbered list) or **unordered** (a bullet-point list).

1. Unlike the some of the elements we have used so far, lists have stricter rules regarding how elements can be nested.

Start a new ordered list underneath the **h3** heading that says **Cover Versions** using an **ol** element:

```
<h3>Cover Versions</h3>
<ol>
</ol>
```

2. Within the **ol** element you can 3 individual list items as **li** elements. Fill each list item with text about cover versions from the Wikipedia article.

```
<ol>
    <li> In 1997, French boy band 2Be3 covered the song under the name "Toujours là
pour toi", which had success in France (No. 4) and Belgium (Wallonia) (No. 12).</li>
    <li> In 1998, an Italian cover entitled "Non ti lascerò" was made by
Fiorello.</li>
    <li>Eurobeat artist Kevin Johnson covered the song in 2004 for the album Super
Eurobeat 149.</li>
</ol>
```

3. Turn the **ordered list** in to an **unordered list** by changing the **ol** elements into **ul** elements:

```
<ul>
    <li> In 1997, French boy band 2Be3 covered the song under the name "Toujours là
pour toi", which had success in France (No. 4) and Belgium (Wallonia) (No. 12).</li>
    <li> In 1998, an Italian cover entitled "Non ti lascerò" was made by
Fiorello.</li>
    <li>Eurobeat artist Kevin Johnson covered the song in 2004 for the album Super
Eurobeat 149.</li>
</ul>
```

4. Try adding a few more list items within the **ul** element so that you can include information about all of the cover versions shown the article.

   When you are finished, you should have 7 list items in a single unordered list.

Inserting images and hyperlinks (linking to other files)

For this part of the exercise, you will need an image file. You can search for an appropriate image using Google Image search. Save the image to your **images** sub-folder with a short descriptive name.

Remember to record the URL of the webpage that you downloaded the image from as well. For this exercise you can reference the webpage by including the URL in your document as a comment:

```
<!-- Image source: https://imgflip.com/memegenerator -->
```

1. Insert you image underneath the **h2** heading by adding an **img** element:

```
<img />
```

   An **img** element is another example of an **empty element**. Like the **br** element you used earlier, this element has been **self-closed**.

2. To make the **img** element display the image that you downloaded, add a **src** attribute within the tag and set a value that is the URL of the image file:

```
<img src="meme.jpg" />
```

   In this example, a short **relative URL** is being used. As long as the image has been saved in the same folder as the HTML document, and the filename is the same, you should be able to reload the page in your browser to see the image.

3. Now try linking a piece of text in your page to another website. For example, the first time **"Rick Astley"** is mentioned in the page, you could link to his official website.

   Locate the text that you want to link and wrap an **a** tag around it:

```
<p><strong>"Never Gonna Give You Up"</strong> is a song recorded by British
singer and songwriter <a>Rick Astley</a>, released as a single on 27 July
1987.</p>
```

4. In the start tag of the **a** element, add an **href** attribute with the **absolute URL** of the website you want to link to:

```
<p><strong>"Never Gonna Give You Up"</strong> is a song recorded by British
singer and songwriter <a href="https://www.rickastley.co.uk/">Rick
Astley</a>, released as a single on 27 July 1987.</p>
```

   Save your document and reload the page in your browser. Make sure the image is displayed correctly and the hyperlink navigates to the correct website when clicked.

Challenge task!

If time allows, create a second HTML file in the same folder and implement the following functionality:

- Create an **a** element in the first page that links to the second page using a **relative URL**.
- The second page should contain a single **img** element that displays a small image.
- Add an **a** element around the image that links directly to a larger version of the image.

If implemented correctly, a user should be able to click hyperlinks to navigate from the first page to the second page, then from the second page to the large image.