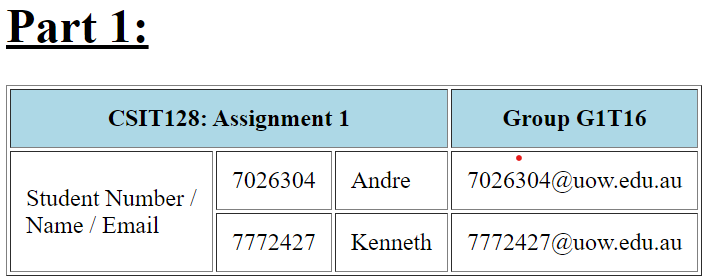
**Our screenshot:**

****

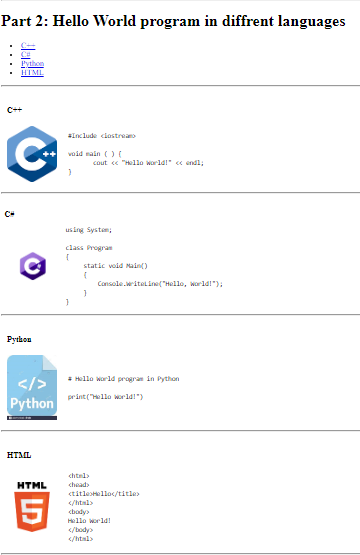
**Our code:**



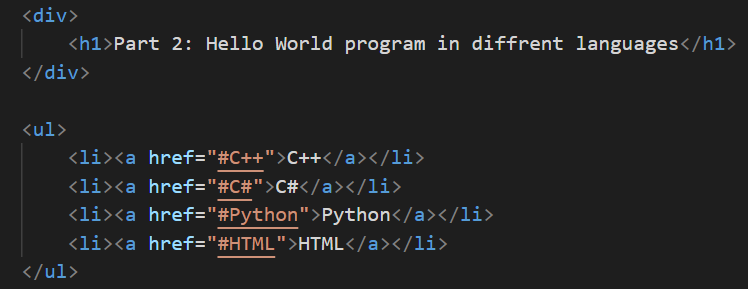
**Explanation:**

Background color of light blue for the table header is set using bgcolor=”lightblue”. The table is styled with borders and padding. As seen, the table header occupies 3 columns of space and the table data cell occupies 2 rows of space.

**Our screenshot:**

****

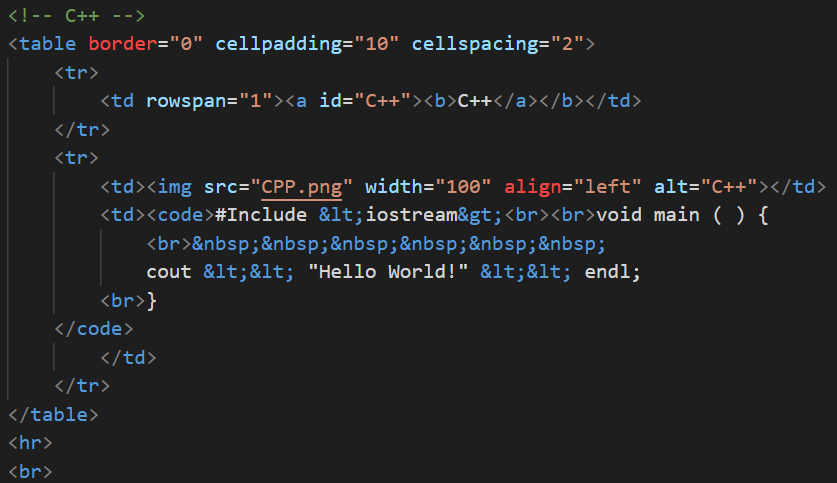
**Our code:**

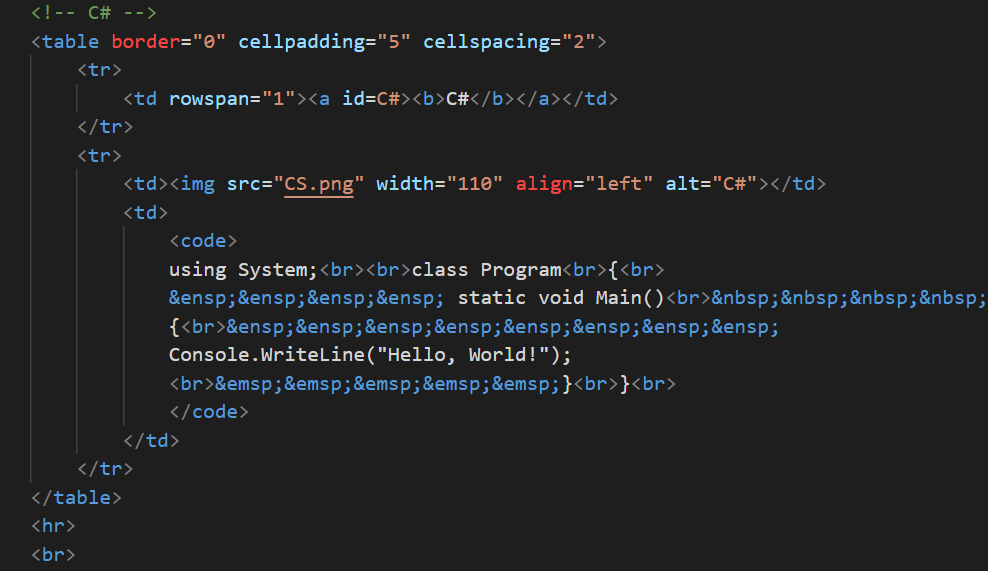
****

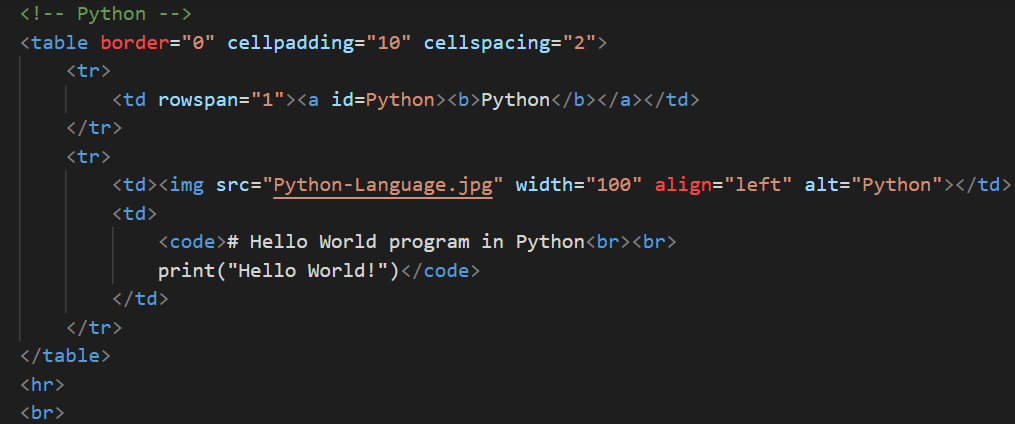
**Explanation:**

Using “<li><a href=...</a><li>” creates an item in the link with a hyperlink. When clicked, it will direct the user to the anchored content within the same page This is done when creating page navigation.

**Our code:**

****

****

****

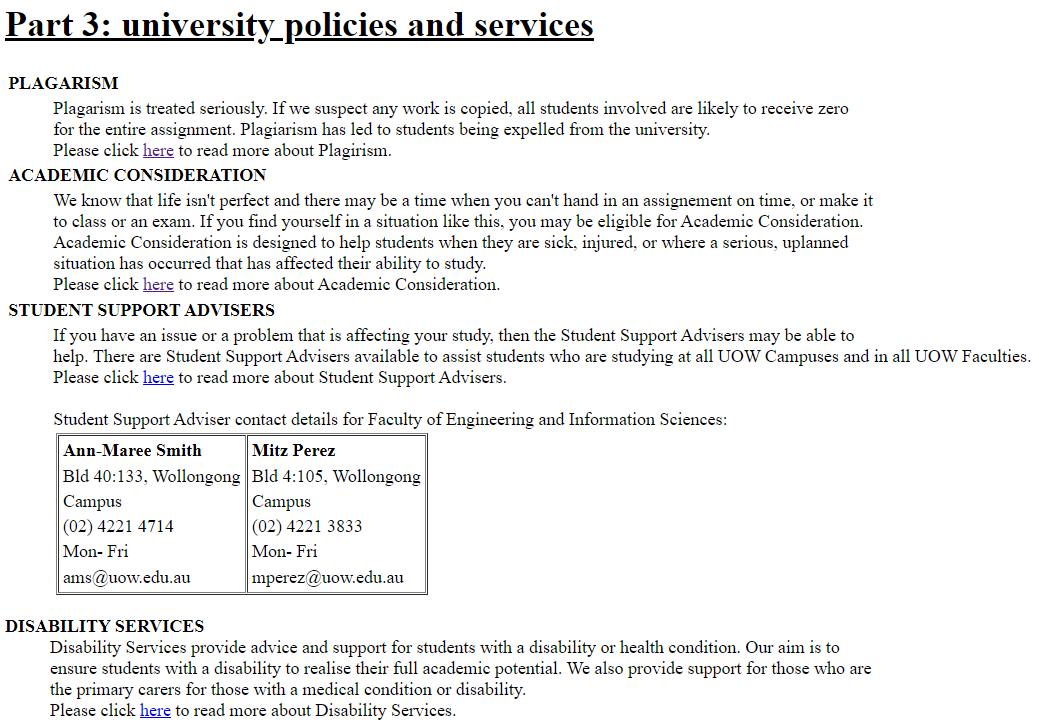
****

**Explanation:**

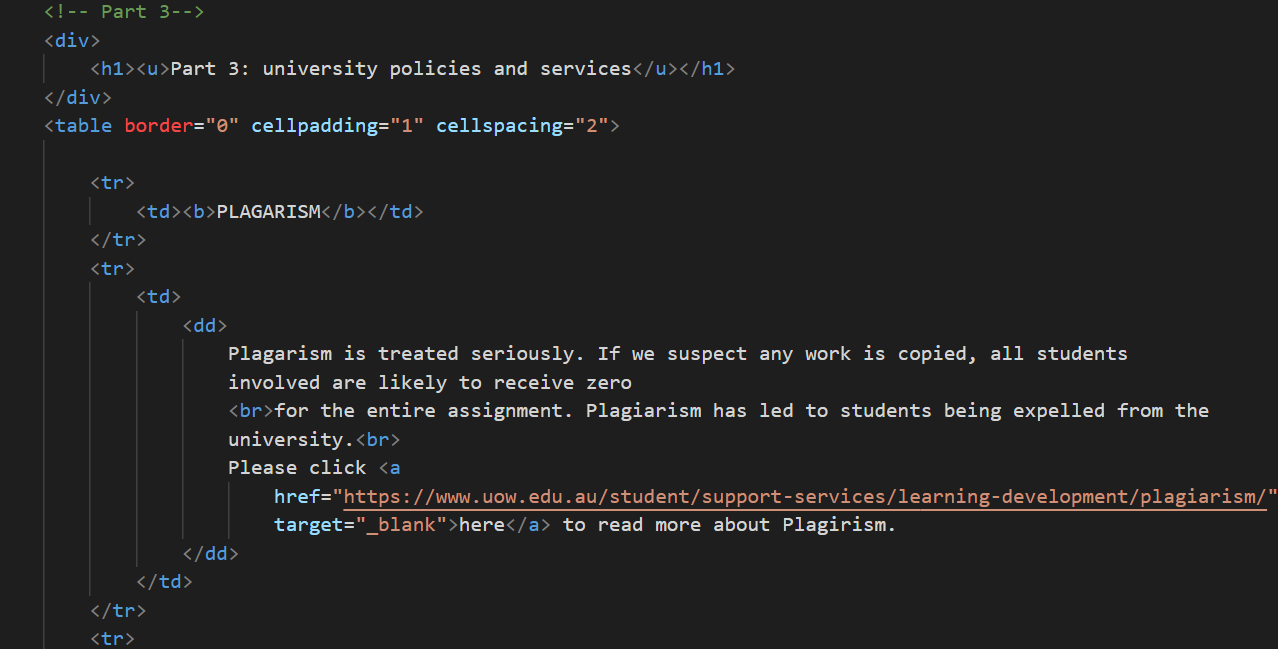
A table with border=“0” was used to make the table lines invisible while making sure that the contents are in each category neatly in its place and a <hr> was used to draw a horizontal line across to mark the end of each item.

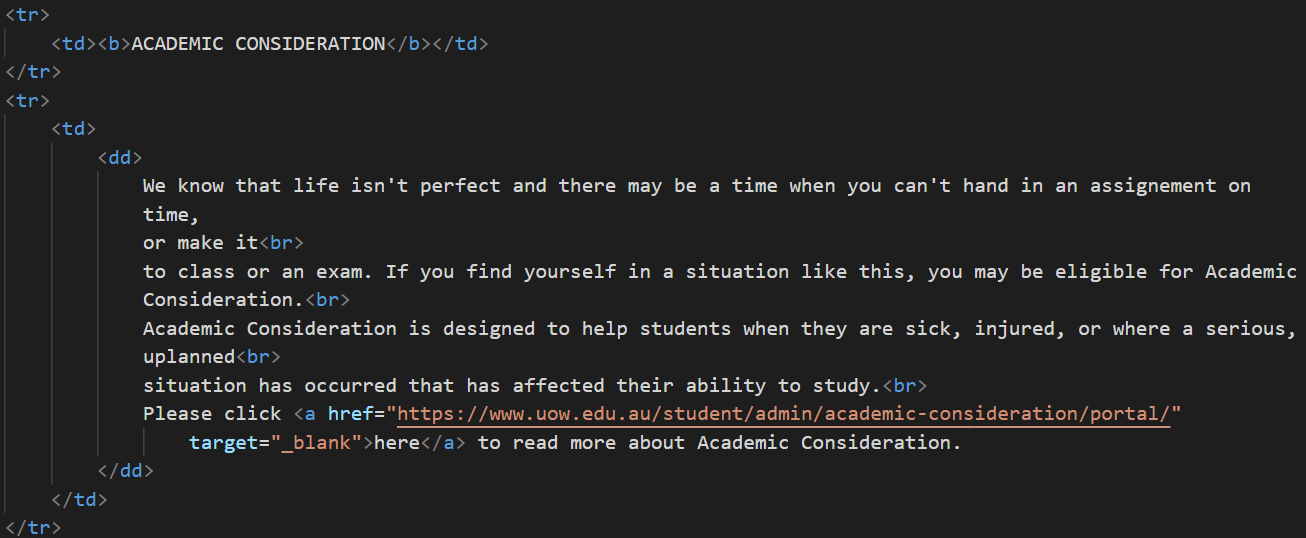
This is used to display code snippets as shown above by the webpage screenshot and code. <code> is used to define a part of text that represents computer code and is often used in code related content in regular text just like the code snippet shown. All have their own ID with image src.

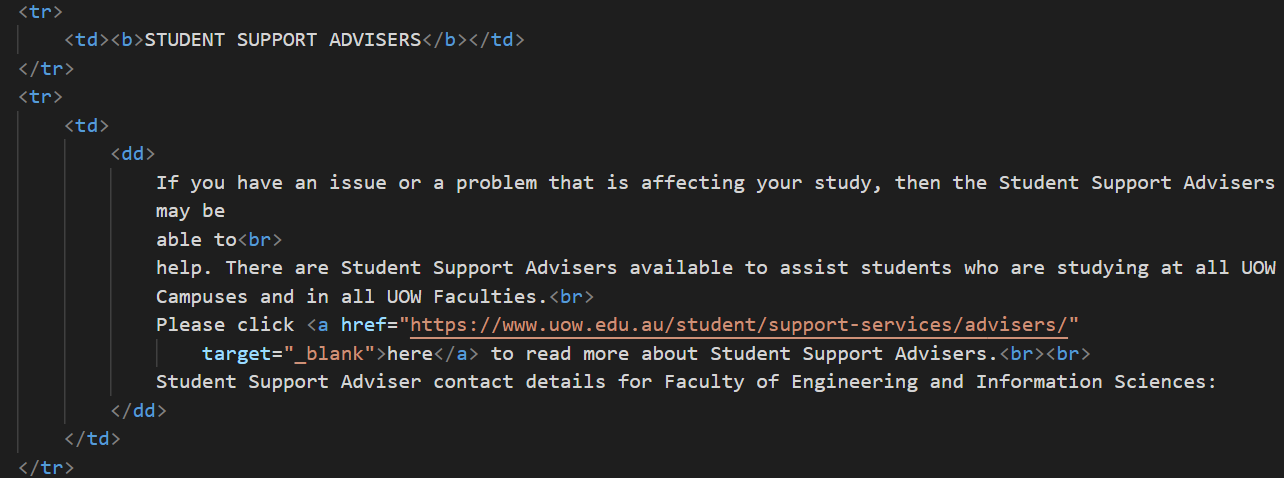
**Our screenshot:**

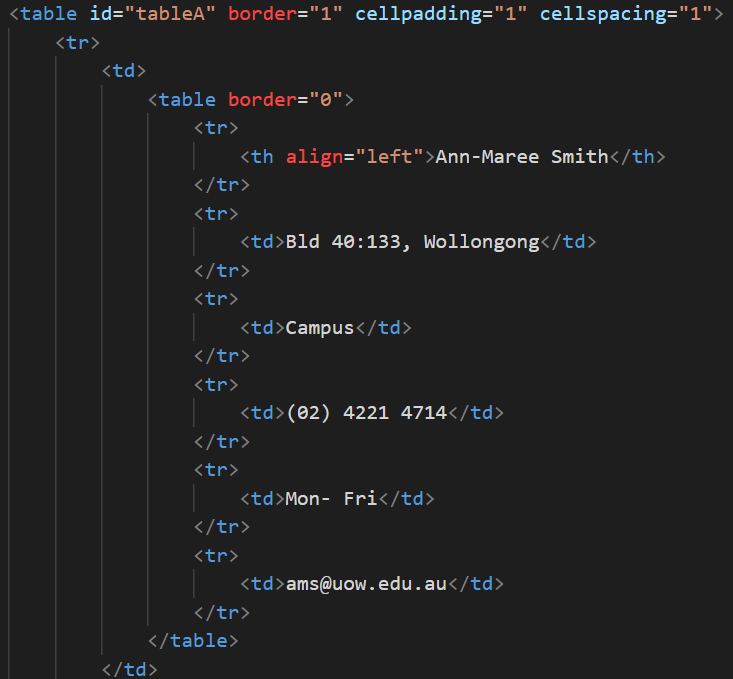
****

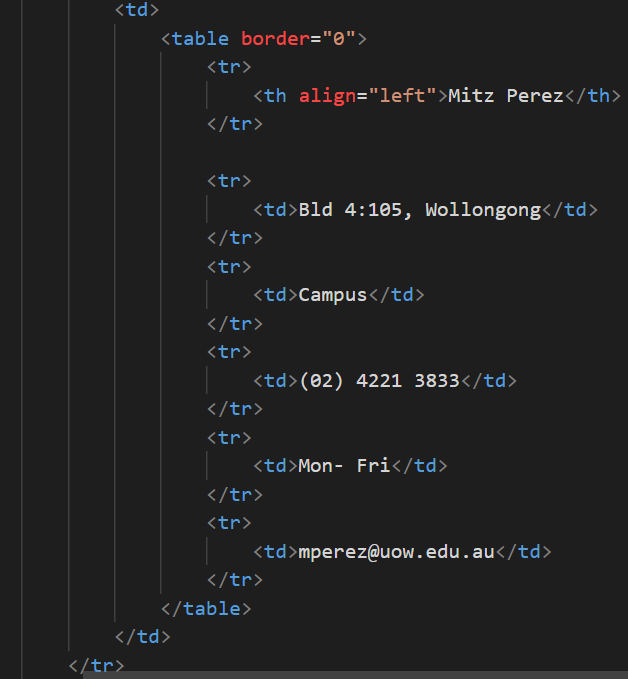
**Our code:**

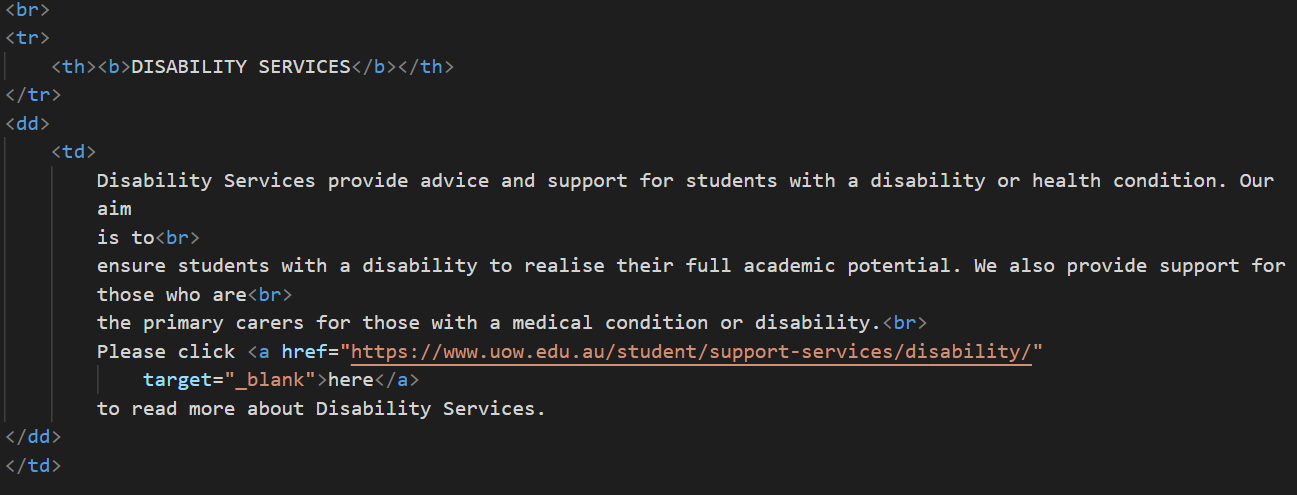
****

****

****

****

****

****

**Explanation:**

<table> was used to make sure everything was aligned neatly. <dd> tag was added in each paragraph to define a description list item, but was mainly used for each paragraph to shift slightly to the right as according to our Assignment 1 specification.

A <a href> was used in between the word ‘here’ in order for it to be hyperlinked to the required website. An extra feature, target=”\_blank” was added while exploring how to open a new page such that the current page would not be directly brought over to another new webpage. This would reduce the frequency of user pressing the back to go back to the previous page in order to look at our Assignment 1 webpage again. In this use case, we believe that it would be a better user experience to add target=”\_blank”.

**Screenshot of name:**

Example 1: Empty

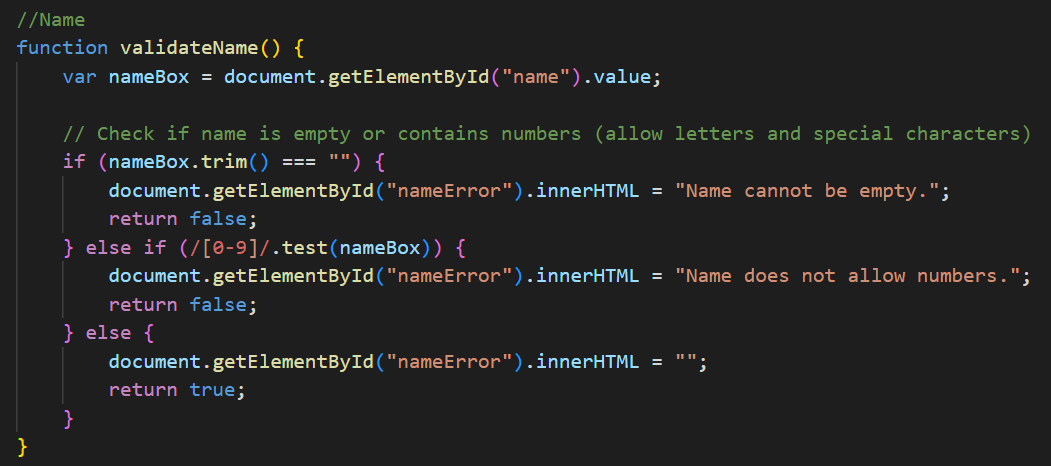


Example 2: Number in name



Example 3: Special Character

**Name code:**

****

****

**Explanation:**

Function validateName() will check for user input of name to match correct format and regular expression is used to achieve it. This function ensures that the input in the ‘name’ text box meets the required criteria such as not being empty nor contain any numbers. If the input does not meet these criteria, the error message will be displayed which means that the function would return ‘false’. Otherwise, it would be ‘true’ which indicates that the input is valid.

**Screenshot of module code:**

Example 1: Left empty



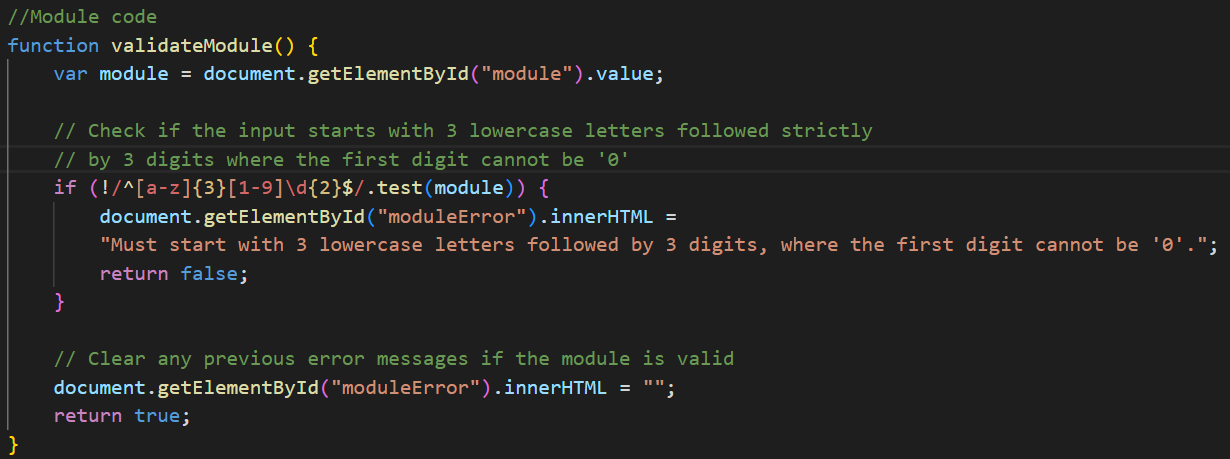
Example 2: Wrong format

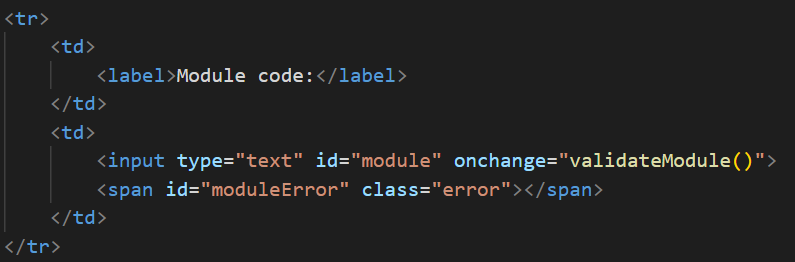


Example 3: Correct format



**Our code:**





**Explanation:**

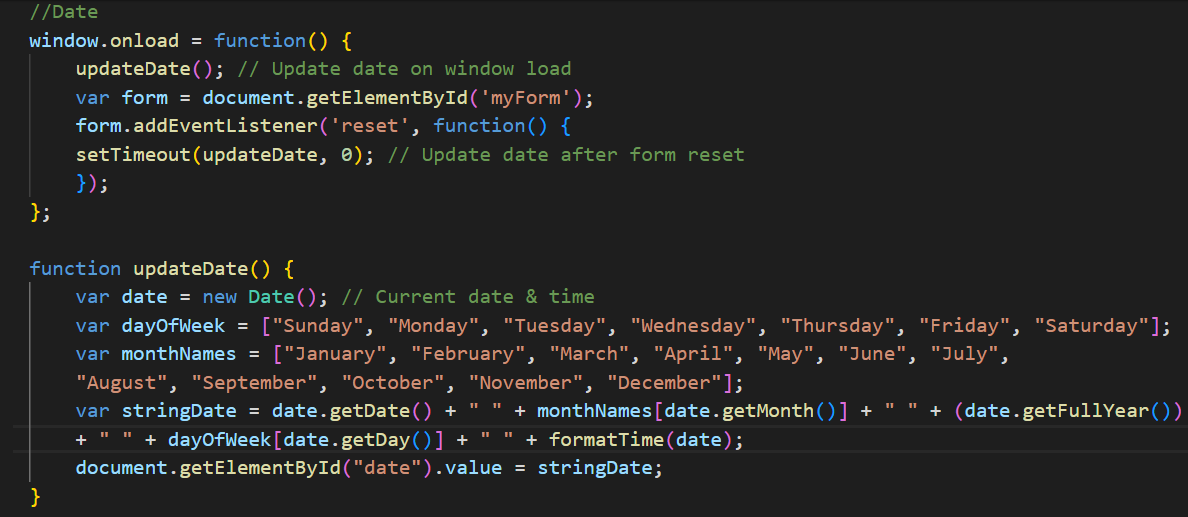
Function validateModule() checks if the user input the correct module code as per format of starting with either left empty or starting with 3 lower alphabets and followed by 3 digits. If input is wrong, an error message is displayed on the side of the “Module code” textbox.

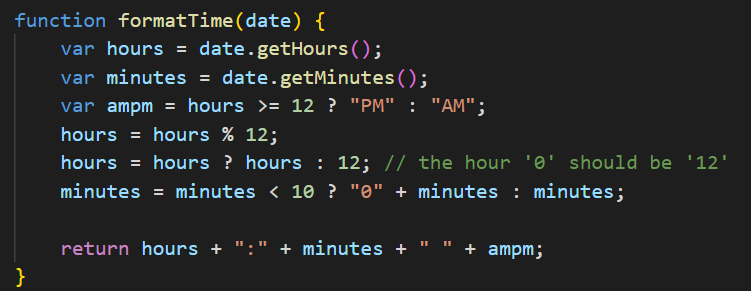
Therefore, this function ensures that the input for the ‘module’ text box follows a specific format. If the input does not meet the required criteria, the error message is displayed accordingly and this means that the function would return a ‘false’. Otherwise, it would return as ‘true’ which indicates that the input is valid.

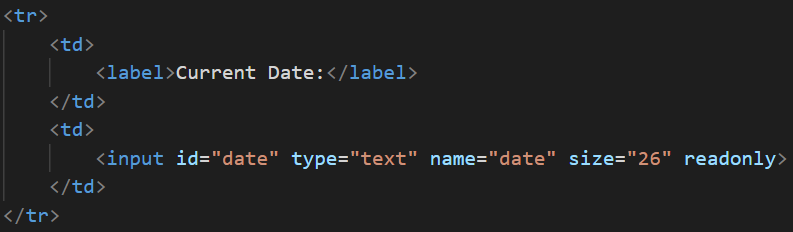
**Screenshot of date:**



**Our code:**







**Explanation:**

This part of the javascript is to display the date and time in a non-editable textbox. It starts out with a ‘window.onload = function()’. This line sets up an even listener that triggers when the webpage loads completely and then it calls the ‘updateDate()’ function to update the date and time. The ‘setTimeout(updateDate, 0)’ inside the reset event listener, this line calls ‘updateDate()’ function using ‘setTimeout()’ with a delay of 0 milliseconds. This effectively schedules updateDate() to run asynchronously after other reset actions are complete.

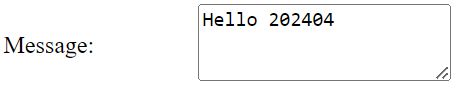
As a whole, this functions ensures that the date and time displayed in the textbox are in the required format and upon loading the webpage, it makes sure that it is always up to date.

**Screenshot message:**

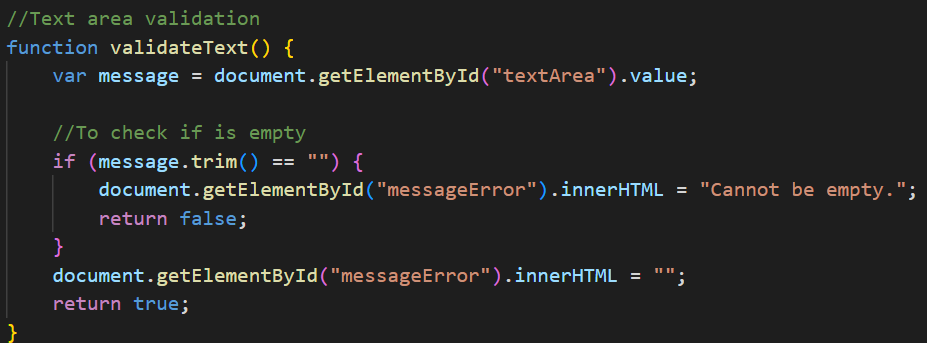
Example 1: cannot be empty



Example 2: on load up with default text



**Our code:**

****

****

**Explanation**

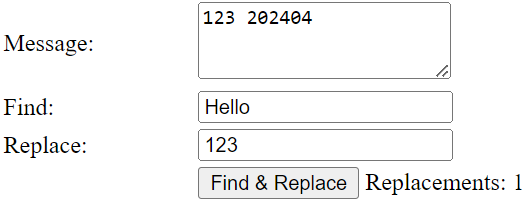
The function validateText() is to designed to validate the user’s input in the name textbox. It checks to make sure that the id of ‘textArea’ is not empty. If it is empty, it will display the error message with the id of ‘messageError’ and returns as ‘false’. Otherwise, it is ‘true’ which means the user’s input is valid.

**Screenshot of Find & Replace**

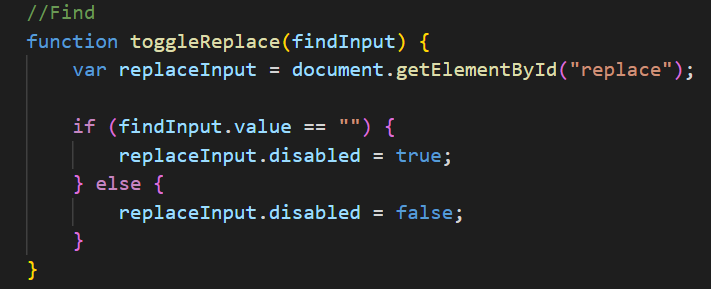
Example 1: Before replace

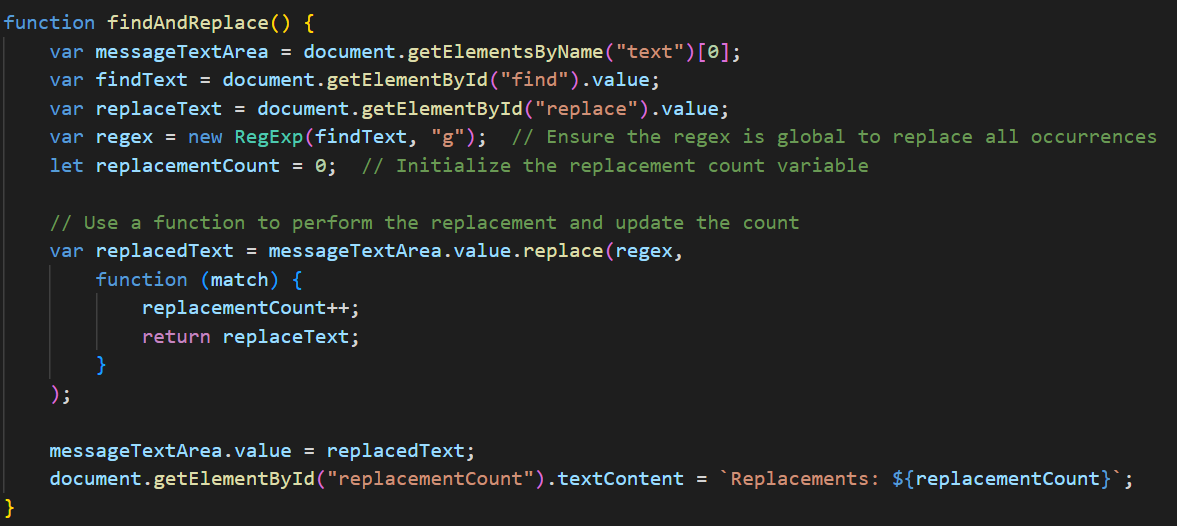


Example 2: After replace



**Our code:**







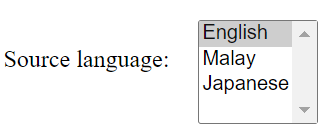
**Explanation:**

The function for ‘toggleReplace(findInput) is called upon when the value of the find input field changes. It takes the findinput as a parameter which represents the input field for the finding text. It then retrieves the replace input field by using ‘getElementById(“replace”) and if the value of the findinput is empty it disables the replace input field, which is done by creating a (‘replaceInput.disabled = true’).

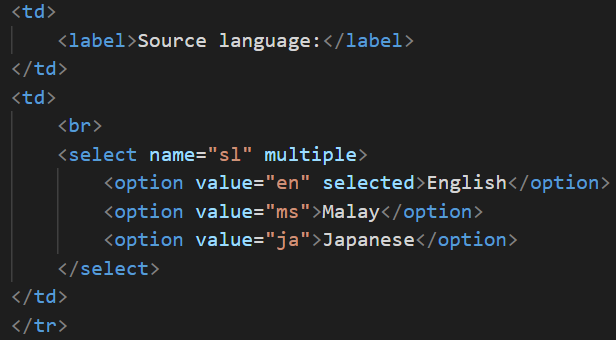
The function for findAndReplace() is called when the user clicks the find and replace button, it would retrieve the text area contents and retrieves its values. It retrieves the values of the find and replace input fields (var findText = document.getElementById("find").value and var replaceText = document.getElementById("replace").value). By creating a regular expression using the findText and sets the "g" flag to ensure global replacement (var regex = new RegExp(findText, "g")). It also initilize a variable ‘replacementCount’ so that it counts the number of replacements made.

The replace() would then replace all occurrences of the findText with the replaceText and each time a replacement occurs, it increments the ‘replacementCount’. The two functions works together such that it enable or disable the replace input field based on whether the find textbox is empty or not. It also find and replace text in the text area while counting the replacements that have been made.

**Screenshot of source language:**

****

**Our code:**

****

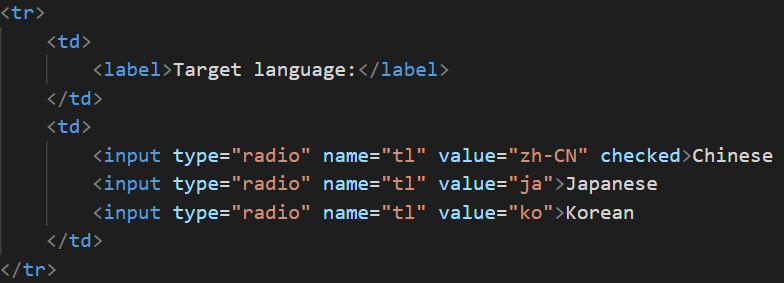
**Explanation:**

This is the list of options for the source language in the HTML body. This displays it in a list such that the user has to choose the options. As the English for this list is required to be the default selected language, a select was included within its line of HTML code ‘<option value="en" selected>English</option>’

**Screenshot of target language:**

****

**Our code:**

****

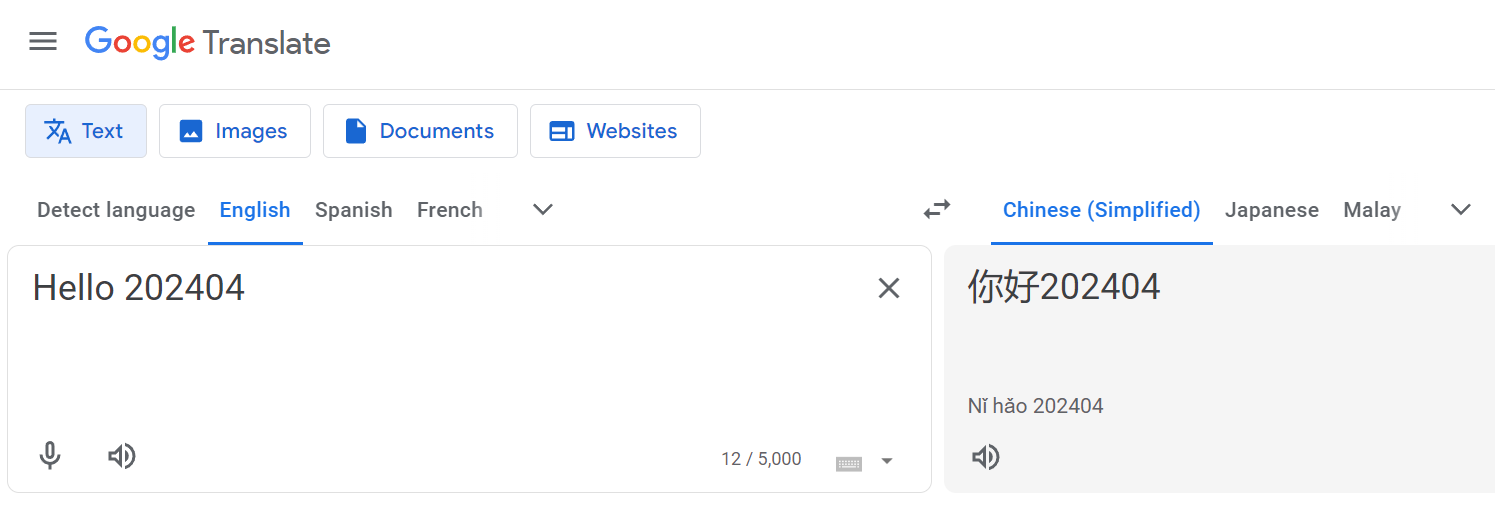
**Explanation:**

Target language selection is by using radio buttons, it displays the different target language and radio selection only allows 1 option at a time.

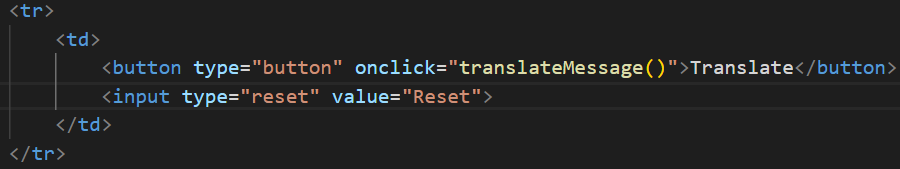
As the chinese language was required to be the default selected language, a ‘checked’ was included in its line of code ‘<input type="radio" name="tl" value="zh-CN" checked>Chinese’.

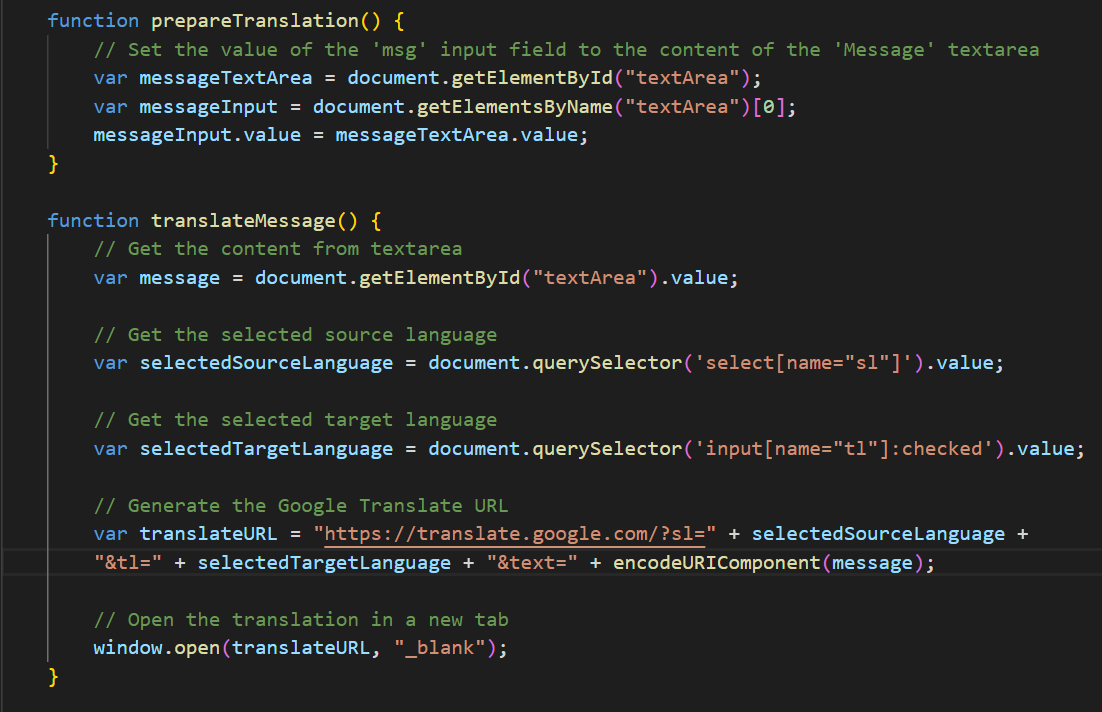
**Screenshot of Translate and Reset**

****

****

**Our code:**

****

****

**Explanation:**

Translate and Reset buttons are created. Using reset is clicked, the form is clear but still will show the default message in the textbox and the current date. Translate will invoke translateMessage() function. The function takes the user input text, recognises the source language and targeted language for translation. It uses the Google translate url, opens a new tab for translation and allows the translation of the selection text easily.

After referencing to the original google translate webpage URL, we realised that in order for the code to google translate the desired language properly, “&tl=” and “&text=” was required and therefore, added into the line for

|  |
| --- |
| var translateURL = "https://translate.google.com/?sl=" + selectedSourceLanguage + "&tl=" + selectedTargetLanguage + "&text=" + encodeURIComponent(message); |