Introduction to Bootstrap

Exercise 1:

Create a basic web page including the CDN for bootstrap from
https://getbootstrap.com/docs/5.3/getting-started/introduction/#quick-start.

2. **Include Bootstrap's CSS and JS.** Place the `<link>` tag in the `<head>` for our CSS, and the `<script>` tag for our JavaScript bundle (including Popper for positioning dropdowns, poppers, and tooltips) before the closing `</body>`. Learn more about our CDN links.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bund
  </body>
</html>
```

Copy the HTML and paste into your html document. Save and run the document to ensure that it works. You should see "Hello World" in a H1 heading.

Add the HTML code shown below.

```
<div class="h1">This is also a H1 heading</div>
```

This should also create a H1 heading as we have specified the H1 class inside the element tag.

Try the following for implementing headings and displays:

```html
<div class="h2">This is a H2 heading</div>
<div class="h3">This is a H3 heading</div>
<div class="h4">This is a H4 heading</div>
<div class="h5">This is a H5 heading</div>
<div class="h6">This is a H6 heading</div>
<div class="display-1">This is a display 1 heading</div>
<div class="display-2">This is a display 2 heading</div>
<div class="display-3">This is a display 3 heading</div>
<div class="display-4">This is a display 4 heading</div>
```

Save and open up your webpage in a browser. Your page should display as follows:

Hello, world!

This is also a H1 heading

This is a H2 heading

This is a H3 heading

This is a H4 heading

This is a H5 heading

This is a H6 heading

This is a display 1 heading

This is a display 2 heading

This is a display 3 heading

This is a display 4 heading

Using the relevant text alignment classes (refer to
https://getbootstrap.com/docs/5.3/utilities/text/#text-alignment), amend your page so the display is
updated as follows:

Hello, world!

This is also a H1 heading

This is a H2 heading

This is a H3 heading

This is a H4 heading

This is a H5 heading

This is a H6 heading

This is a display 1 heading

Using the relevant text colouring classes (refer to
https://getbootstrap.com/docs/5.3/utilities/colors/#colors), amend your page so the display is

updated as follows:

Using the relevant classes for colouring backgrounds (refer to
https://getbootstrap.com/docs/5.3/utilities/background/#background-color), amend
your code so that the following is output:

Exercise 2:

In this exercise, just follow the steps below to put together a 4-page website using bootstrap:

1. Copy a starter template from bootstrap – as you did in the previous exercise. Save your page as page1.html.
2. Add a container div to the body of your webpage. All the elements you will add to this page will be included inside this div element – refer to https://getbootstrap.com/docs/5.3/layout/containers/#default-container.
See screenshot below:

```
<div class="container">
  <!-- Content here -->
</div>
```

3. Within the container div another div. See screenshot below – line 4:

```
3  <div class="container">
4      <div>Bootstrap Carousel</div>
5  </div>
```

4. Assign the following classes to the class attribute of the div element you created in the previous step:
   a. p-4
   b. text-center
   c. fs-1
   d. text-white
   e. bg-primary
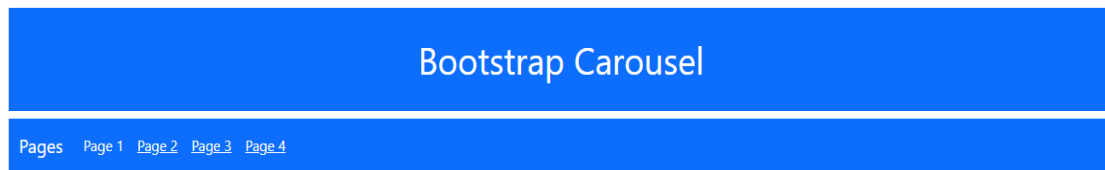   f. mt-2

   Note: The p-4 class handles padding (https://getbootstrap.com/docs/5.3/utilities/spacing/#margin-and-padding), text-center(https://getbootstrap.com/docs/5.3/utilities/text/#text-alignment) centres the text inside the div, fs-1(https://getbootstrap.com/docs/5.3/utilities/text/#font-size) applies the font size to the text inside the div, text-white(https://getbootstrap.com/docs/5.3/utilities/colors/#colors) applies the font colour, bg-primary(https://getbootstrap.com/docs/5.3/utilities/background/#background-color) applies a background colour to the div, while mt-2(https://getbootstrap.com/docs/5.3/utilities/spacing/#margin-and-padding) applies a top margin to the div.

   Please read up on each of these classes by following the links in brackets to see what other values you can use to apply different values for padding, margin-top, font-size etc …

   Once you have applied these classes to the div, save your html page and open it in a browser, you should see the following:

Bootstrap Carousel

5. Underneath this div, add a nav component (refer to https://getbootstrap.com/docs/5.3/components/navbar/#nav), and amend it so your page now looks as follows:

Note, you will need to apply some bootstrap classes to the nav component to change the font colour on the links to white, the background colour to the same colour as the first div – refer to previous step, and margin top to the same value as the first div – again refer to previous step.

The screenshot above shows an underline on the Page 2, Page 3, and Page 4 links. You can use the text-decoration-underline class to set this – please refer to https://getbootstrap.com/docs/5.3/utilities/text/#text-decoration for more information on this class. Note, Page 1 does not have the underline as it's the current page.

6. Save all your changes and then copy page1.html to page2.html, page3.html and page4.html.
7. Update the links in the nav component for each page so that the correct links have the text-decoration-underline class and the href values are configured to link between the pages.
8. Test you can navigate between the pages using the nav component links.
9. On page1.html add a carousel component just underneath the nav component. You can copy a template for the carousel component from bootstrap - refer to https://getbootstrap.com/docs/5.3/components/carousel/#autoplaying-carousels.
10. Note that the carousel has 3 images. You can amend the src attributes on each of these to refer to the images which are included along with this lab in the same section on Moodle – see lab_images.
11. Just download the lab_images.zip file and extract the images from it – make sure to copy the images to the same folder location as your page1.html file and don't forget to update the src attributes on the img tags with the image filenames e.g. src="elephant.jpg"
12. Apply the mt-2 to the carousel in order to set the margin top.
13. Save all your changes and open page1.html in a browser. You should see something like the screenshot on the following page:

14. Test out the button controls on the carousel < and > to navigate between the 3 different images. As it is an auto play carousel, the images should change at periodic intervals anyway.
15. On page2.html add a button which will launch a Modal component just underneath the nav component. See https://getbootstrap.com/docs/5.3/components/modal/#live-demo.
16. Apply the mt-2 class to this button to set the margin-top property.
17. Save your changes and open page2.html in a browser. Your page should look as follows:

18. It is optional to modify the button label, but do click on it and you should get the following pop up:

19. It is also optional to update the text values in the Modal component.

20. On page3.html add a h2 element and a table (see
    https://getbootstrap.com/docs/5.3/content/tables/) just underneath the nav component.
21. Make sure the h2 (with text "CSS Frameworks Table") and table are styled in a similar way to
    what is shown in the screenshot below:

22. On page4.html add a live alert (see
    https://getbootstrap.com/docs/5.3/components/alerts/#live-example) just underneath the
    nav component.
23. Apply the mt-2 class to the button and div. You will require the JavaScript code included with
    the live example to ensure the button click works to produce an alert. The JavaScript is
    available just underneath the live example when you open the link from step 22.
24. Make sure to fully test the alert is presented when the button is clicked and that the alert can
    be dismissed.
25. Make sure to fully test your four pages, they are styled appropriately(refer to screenshots) and
    ensure navigation is possible between each page.

Exercise 3:

In this exercise, you will use JavaScript to perform client-side validation on a user registration
page.

Use the Bootstrap Form Components to complete this exercise:
https://getbootstrap.com/docs/5.3/forms/overview/

Create a HTML page (register.html) like the one below:

Registration Page

Enter your details in the form below to create an account.

Username

Username

First Name

First Name

Last Name

Last Name

Email address

Email

Password

Password

Confirm Password

Password

Submit

You should create a validation function to validate each input field. The criteria for validation are:

Username: at least 8 characters long

First Name and Last Name: cannot be empty

Email address: must contain an "@" symbol and a full stop "."

Password: Must be at least 5 characters long

Confirm Password: Must match the input in the password field

If validation fails for any form element, an alert box should appear giving details of the error.
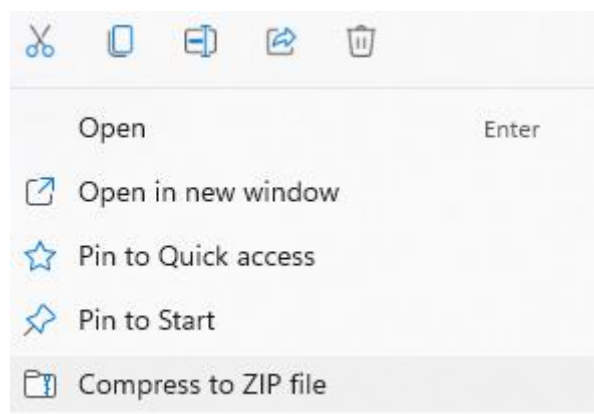
Use the form's onsubmit event attribute to call your validation function – see the following page at w3schools:

JavaScript Form Validation (w3schools.com)

Hints: The length property, and the includes() and/or indexOf() functions will be useful.

Upload your work to Moodle

1. Navigate to the location of the folder where you saved all your work for today's lab.
2. Right-click on the folder and select "Compress to ZIP file". This will create a compressed version of any files you have worked on for the lab.

3. There should be a new compressed file created. This is the file that you will need to upload to Moodle.
4. In Moodle, navigate to the "Submissions" section, and click the ***upload*** link for current lab.
5. Click "add submission" and add the ZIP file you created here. Make sure you complete the submission process. Your lab work has been submitted.