Colour and Fonts in CSS

IMPORTANT! Save all your work to a safe location such as OneDrive.

Create a folder for GUI & Web Development into which you will save all your work for this module, arranged how you wish. Ideally you should create a folder each week for your lab exercises. Note that you should create **a separate file** for each exercise.

This week we will use CSS to change colors to elements, change the styles applied to type, and use different fonts in web pages. In addition to this, we will take a look at pseudo classes in CSS and see how they can be used to add some interactivity to a web page.

Exercise 1: Exploring different ways of specifying colors

1. Create the following page that has a single H1 heading as shown below:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"/>
    <title>Exercise 1</title>
    <style>
        h1 {color: red;}
    </style>
</head>
<body>
    <h1>This is a foreground colour!</h1>
</body>
</html>
```

2. Save your page as a HTML document and view it in a browser. There should be a single h1 heading coloured red.

This is a foreground colour!

This h1 has the colour specified using a *named* colour – in this case red

3. Change the rule in the CSS for this colour, so that the colour is specified using the RGB model, as shown below:

```
<style>
    h1 {color: rgb(255, 0, 0);}
</style>
```

Save and refresh your page. The same colour should still be displayed.

4. Change the rule in the CSS for this colour, so that the colour is specified using the Hex model, as shown below:

```
<style>
    h1 {color: #ff0000;}
</style>
```

Save and refresh your page. The same colour should still be displayed.

5. The RGB colour model uses three colours – red, green and blue (rgb) – to create a range of colours. The possible values range from 0 to 255 (integers only, no decimals) for each colour. For example, to specify blue in the rgb model, use the values 0 for red, 0 for green, and 255 for blue – for example:

```
color: rgb(0, 0, 255)
```

RGB

RGB defines how much red, green or blue value to display in a decimal integer value somewhere between 0, which is no representation of the color, and 255, the highest possible concentration of the color. So, in the example rgb(255, 0, 0), you get a very bright red. If you wanted all green, the RGB would be rgb(0, 255, 0). For a simple blue, it would be rgb(0, 0, 255).

Hexadecimal

Hexa means six (in Greek) and ***decimal*** (from the Latin for 10) means ten, which stands for the base 16 number system. You'll often hear it just referred to as "Hex". In HTML and CSS, Hexadecimal colours start with a hashtag, and are followed by six letters or numbers. The first two digits refer to red, the next two to green, and the last two are blue. You'll define how much red, green, and blue in values between 00 (which is zero) and FF (which is 255) instead from 0 to 255 like in RGB. A colour that's maximum red, no green, and no blue would be #FF0000.

The following colour in rgb:

```
h1 {color: rgb(0, 255, 255);}
```

is the same as shown below in hex:

```
h1 {color: #00FFFF;}
```

Exercise 2: Using External Fonts.

1. Create a basic HTML page with a single H1 Heading and a paragraph as shown below:

```
<!DOCTYPE html>
<html>
<head>
    <title>Exercise 2</title>
</head>
<body>
    <h1>This is a H1 Heading</h1>
    <p>This is a paragraph</p>
</body>
</html>
```

2. Change the font used for the paragraph to “Sacramento”. You can use **google fonts** to create the link and associated style code for this.

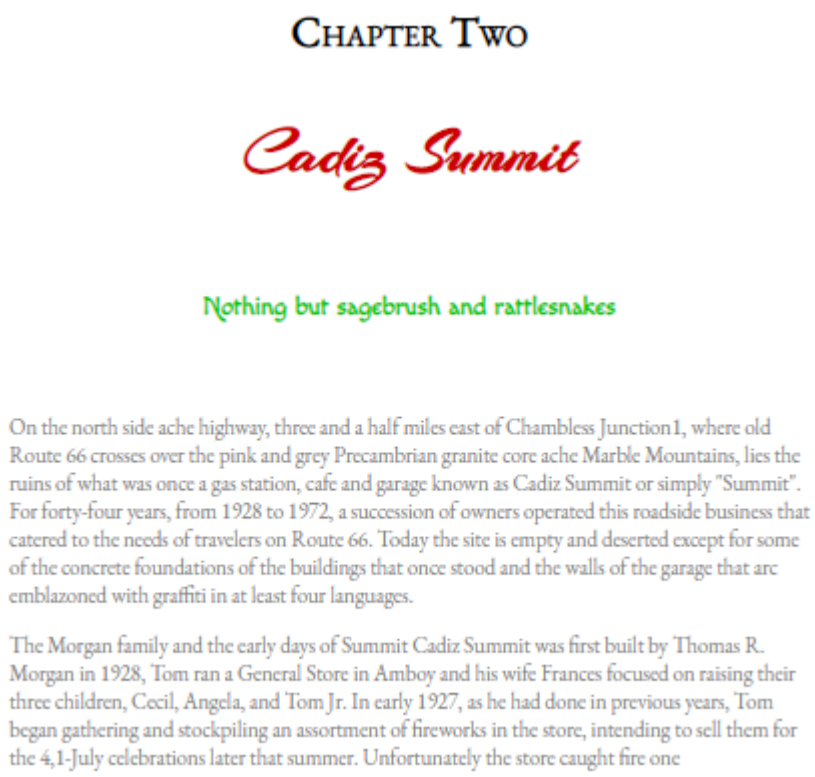
3. The font size may be enlarged to 200% for the paragraph using the rule shown below, which should make it more legible.

```
p { font-family: 'Sacramento', cursive;
    font-size: 200%;
}
```

4. Specify a size of 200% for the H1 heading. Note that this will not look any different on your browser as the default font size for H1 is 200% (on PC browsers) of the default size (so it’s already using this scale). Change the size to 300%, save and run the page again. The H1 heading should now be larger.

Exercise 3: Using Multiple External Fonts.

1. In this exercise you will create a page similar in style to the following:



2. The HTML for this page is as follows:

```
<h1>Chapter Two</h1>
<h2>Cadiz Summit</h2>
</br>
<h3>Nothing but sagebrush and rattlesnakes</h3>
</br>
</br>
<p>On the north side ache highway, three and a half miles east of
Chambless Junction1, where old Route 66 crosses over the pink and grey
Precambrian granite core ache Marble Mountains, lies the ruins of what was
once a gas station, cafe and garage known as Cadiz Summit or simply
"Summit". For forty-four years, from 1928 to 1972, a succession of owners
operated this roadside business that catered to the needs of travelers on
Route 66. Today the site is empty and deserted except for some of the
concrete foundations of the buildings that once stood and the walls of the
garage that arc emblazoned with graffiti in at least four languages.</p>
<p>The Morgan family and the early days of Summit Cadiz Summit was first
built by Thomas R. Morgan in 1928, Tom ran a General Store in Amboy and
his wife Frances focused on raising their three children, Cecil, Angela,
and Tom Jr. In early 1927, as he had done in previous years, Tom began
gathering and stockpiling an assortment of fireworks in the store,
intending to sell them for the 4,1-July celebrations later that summer.
Unfortunately the store caught fire one</p>
```

   Just copy this html into a web page.

3. All your CSS should be added to an *external* style sheet called *style.css.*

4. Fonts used are Garamond, Arizonia, and Macondo Swash Caps. Use google fonts to add your fonts to this page.

5. The HTML for this page should be nested inside a div element:

6. Apply the following rule for the div element:

```
#main {
        margin: auto;
        width: 600px;
}
```

   This rule ensures that the text fits in a “div” that is 600px wide, and is centred on the page.

7. Specify all colours using RGB.

Exercise 4: Create CSS Logo

1. Recreate the following logo using HTML and CSS:

2. Fonts to use in this exercise are the following:
   a. Sigmar One
   b. Caveat

3. Use the span element to target the different parts of the text for styling. See HTML span tag (w3schools.com).

Exercise 5: Create Multi-coloured logo

1. Change the output from exercise 4 to as shown below:

Exercise 6: Using Pseudo Classes

1. Create the page shown below that is using a Pseudo class. Note that this class is using the hover pseudo class to change the colour of the heading when the mouse hovers over it.

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Pseudo Classes</title>
    <style>
        h1:hover {color: red;}
    </style>
</head>
<body>
    <h1>Hello Pseudo!</h1>
</body>
</html>
```

2. Add an additional 3 headings to your page, as shown in screenshot on the next page. Save your page and view it. Each header should change colour as the mouse hovers over it.

```
10  <body>
11      <h1>Hello Pseudo!</h1>
12      <h1>Heading 2</h1>
13      <h1>Heading 3</h1>
14      <h1>Heading 4</h1>
15
16  </body>
17  </html>
```

3. Amend you code so that "Heading 4" does not change colour when the mouse hovers over it, but all other H1 heading do.

Exercise 7: Using Pseudo Classes on links

1. Create a page with a link to the following page:

   "https://www.w3.org/People/Berners-Lee/1996/ppf.html"

   Add a background colour of dark blue to the page using CSS, and change the link so that:

   - It does not have an underline
   - Its colour is red (not the default blue)
   - On hover, the colour changes to orange
   - When it is clicked (active), it changes to green
   - Visited links should be grey

   Note that all colours should be specified using **hexadecimal**.

Exercise 8: Shadows

1. Create a page with four Level 1 headings as per screenshot below:

   Text Shadow North East Black

   Text Shadow North West Blue

   Text Shadow South East Red

   Text Shadow South West Green

2. You task is to create text shadow CSS rules for each of the level one headings so that the

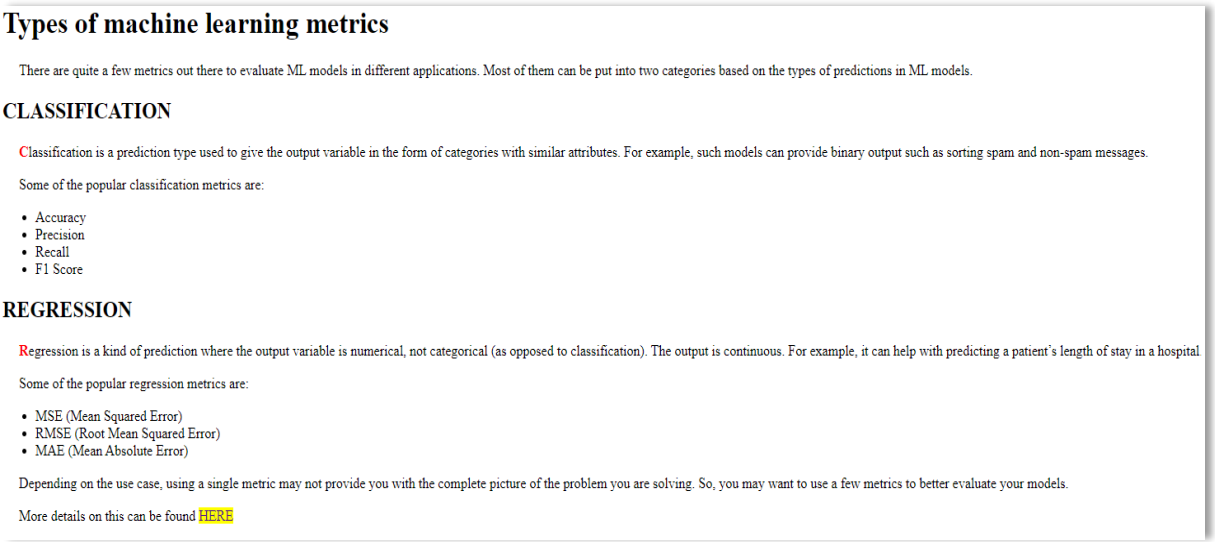
shadow casts in the direction as indicated by the text in each heading, and the shadow color is as indicated in the text. Once the CSS rules are applied your page should look as follows:

Note: Refer to slides 36 and 37 for help with this exercise. set the blur value for each to 5px.

Exercise 9: More CSS Rules

1. Download machine_learning_metrics from the lab 8 folder on Moodle.
2. Create CSS Rules to do the following:
   - Change the text in any level heading 2 to uppercase
   - Indent each paragraph by 20px.
   - In the second and fourth paragraph, change the font-weight of the first letter to bold and color red.
   - In the first unordered List, update the first letter of each list item to uppercase.
   - In the second unordered List, update the initials to uppercase e.g. mae becomes MAE.
   - Modify the hyperlink so the underline is removed, the text is transformed to uppercase and the background color is yellow.

On completion, your page should look as follows:



Types of machine learning metrics

There are quite a few metrics out there to evaluate ML models in different applications. Most of them can be put into two categories based on the types of predictions in ML models.

CLASSIFICATION

Classification is a prediction type used to give the output variable in the form of categories with similar attributes. For example, such models can provide binary output such as sorting spam and non-spam messages.

Some of the popular classification metrics are:

- Accuracy
- Precision
- Recall
- F1 Score

REGRESSION

Regression is a kind of prediction where the output variable is numerical, not categorical (as opposed to classification). The output is continuous. For example, it can help with predicting a patient's length of stay in a hospital

Some of the popular regression metrics are:

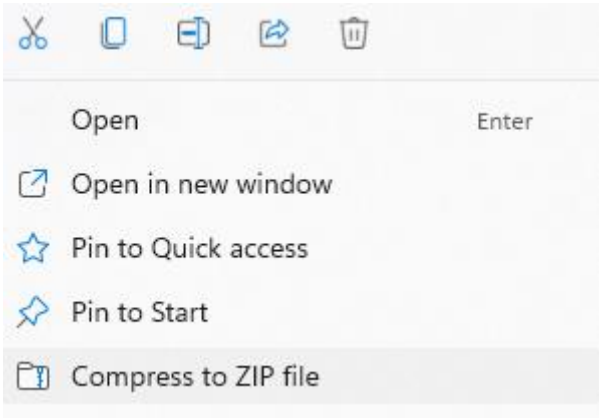
- MSE (Mean Squared Error)
- RMSE (Root Mean Squared Error)
- MAE (Mean Absolute Error)

Depending on the use case, using a single metric may not provide you with the complete picture of the problem you are solving. So, you may want to use a few metrics to better evaluate your models.

More details on this can be found HERE

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.