

# Exercise Week 7

---

## Using the DOM

### Exercise 1

1. Create a web page that has a heading and a button as shown below.

Change text

**This is JavaScript**

When the “change text” button is clicked, the heading should change to text as shown below:

Change text

**This is DOM manipulation**

### Exercise 2 - Basic Calculator

1. Create a web page that has 2 input boxes and a button as shown below.

*Your page should allow the user to input 2 numbers and clicking the add button will create an alert with the answer ( $num1 + num2$ ) using `getElementById` and a function.*

**Please note the following:**

The default value type from an input box is a string (text). This means that if you try and “add” the values from the first and second box it will concatenate them – eg:  $5 + 5 = 55$

You will therefore need to convert the type to a number. See the example below:

```
var num1 = Number(document.getElementById("inputBox1").value);
```

This is creating a variable called `num1`, that is a number, and contains the value inside a HTML element with an ID of “inputBox1”. This will ensure that the variable ***num1*** is a number

### Exercise 3 - Calculator

1. Extend the calculator created in exercise 2 so that it is similar to the page below:

**Calculator**

Number 1:

Number 2:

**ANSWER: 6 divided by 2 = 3**

Buttons should add, subtract, divide and multiply accordingly.

## Functions in JavaScript

### Exercise 4 - Guessing Game 1 to 10

Write a JavaScript program where the program generates a random number between 1 to 10 . The user is then prompted to input a number. If the number the user input matches with the random number, the program will display a message "Good Guess", otherwise display a message "Sorry, wrong number".

How to generate a random number between 1 and **10** in JavaScript:

```
var num = Math.ceil(Math.random() * 10);
```

### Exercise 5 - Guessing Game 1 to 100

Write a JavaScript Guessing Game program where the program chooses a random number between 1 to 100. The user is then *prompted* to guess the number the program chose. The user has six attempts to guess the right number. After each attempt, if the number is incorrect, the program will display "Guess too high" or "Guess too low" depending on the user input. If the user has not guessed the correct number after 6 attempts, the game ends and the number the program generated is displayed.