



Introduction to JavaScript Lab

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.

In this week's lab, we will begin using JavaScript. When you have completed all exercises, upload them to the link provided in Moodle.

Exercise 1: Hello World!

1. Create a simple hello world application that pops up a message saying "Hello World!" when the page is opened.

```
1 <html>
2 <head>
3   <title>My Web Page</title>
4 </head>
5 <body>
6 </body>
7 </html>
8 <script>
9   alert("Hello World!");
10</script>
```

Amend your page so that it also displays a welcome message on the web page.

```
1 <html>
2 <head>
3   <title>My Web Page</title>
4 </head>
5 <body>
6 </body>
7 </html>
8 <script>
9   alert("Hello World!");
10  document.write("Welcome!");
11</script>
```

Exercise 2: Variables in JavaScript

1. Create a variable containing **your name** and output the value to the page. Add comments to explain your code.

```
1 <html>
2 <head>
3   <title>My Web Page</title>
4 </head>
5 <body>
6 </body>
7 </html>
8 <script>
9   //create a variable called myVar
10  let myVar = "Mike";
11
12  //output the variable to the page
13  document.write(myVar);
14 </script>
```

Amend your code so that your name is also displayed using an alert.

2. Create 2 variables, called ***num1*** and ***num2***. Output the addition of these 2 numbers to the page using an alert.

```
1 <html>
2 <head>
3   <title>My Web Page</title>
4 </head>
5 <body>
6 </body>
7 </html>
8 <script>
9   //create 2 variables
10  let num1 = 10;
11  let num2 = 20;
12
13
14  //output addition to the page
15  alert(num1 + num2);
16
17 </script>
```

3. Amend your page so that the following operators are used, and the results are output:

```
//output
alert(num1 + num2);
alert(num1 - num2);
alert(num1 * num2);
alert(num1 / num2);
alert(num1 % num2);
alert(num1++);
alert(num1++);
```

4. Change your code so that your results are output to the page:

```
//output
document.write(num1 + num2);
document.write(num1 - num2);
document.write(num1 * num2);
document.write(num1 / num2);
document.write(num1 % num2);
document.write(num1++);
document.write(num1++);
```

Note that the results are output in a single line across the page.

HTML tags can be added to the *document.write* to ensure that **each result** is output on a new line – for example:

```
document.write(num1 + num2 + "<br>");
```

Note that JavaScript will concatenate the `"
"` to the output and the browser will interpret this as HTML.

Exercise 3: Input using prompt()

1. In this exercise, you will create a webpage that when opened, prompts (with a pop up) the user to enter their name.

To do this, you will *prompt* or ask the user to input their name. You will use the `prompt` method to do this. The `prompt()` method displays a dialog box that prompts the visitor for input, and has the following format:

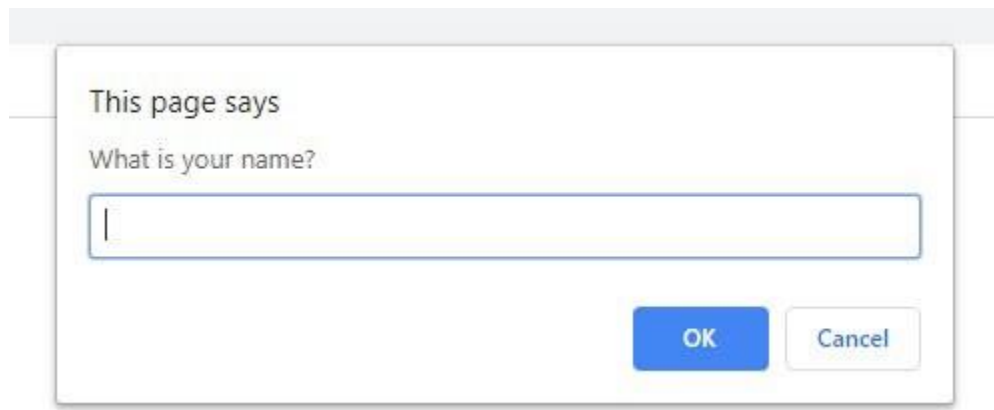
One method of getting a user to input information is by using the JavaScript ***prompt*** command. The `prompt` command has the following structure:

```
prompt("What is your name?");
```

Enter the JavaScript code above in a new page as shown:

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title> </title>
6 </head>
7 <body>
8 </body>
9 </html>
10
11 <script>
12
13     prompt("What is your name?");
14
15 </script>
```

After running this page, a pop-up window appears in your browser window with a text field, as shown on the next page:



Enter your name and click OK. The pop-up window disappears, and your page is blank.

To store the information entered by the user, you need to store it in a variable.

To store user-entered data in a variable, you create a new variable and then follow it with `=`. You then follow it with the prompt statement, as shown here:

```
let username = prompt("What is your name?");
```

When this code executes, the prompt will run first, asking the user to enter data. Once the OK is pressed, then that data will be saved into the variable – in this case, called `username`.

Now you can output the information to the page using the variable you created, for example:

```
document.write(username);
```

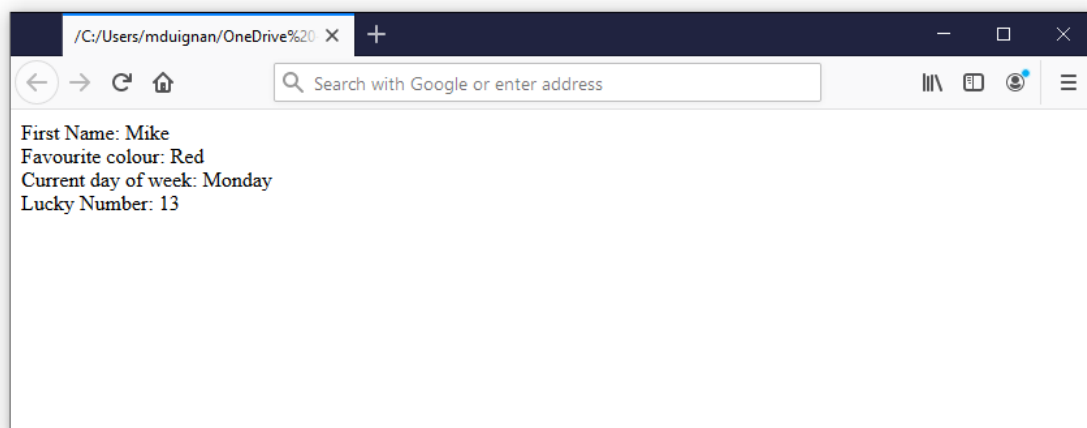
Exercise 4: Multiple Inputs

1. In this exercise, you will create a webpage that when opened, prompts the user (with a pop up) to enter multiple values using multiple prompts.

Implement prompts to ask the user for the following information:

- First Name
- Favourite colour
- Current day of week
- Lucky Number

Your finished page should produce the output as shown below:



2. Amend your code so that the user receives a greeting along with the name (using concatenate):

```
document.write("Welcome " + myName);
```

Exercise 5: Basic Calculator

1. Create a simple calculator that prompts a user to input 2 numbers, adds them and displays the result in the web page.

```
let num1 = prompt("Enter first number:");  
let num2 = prompt("Enter second number");  
  
document.write("Result: " + (num1 + num2));
```

Note the problem with this. Your result is not the addition of the 2 numbers, but rather the concatenation of 2 strings.

IMPORTANT!

JavaScript uses the plus symbol for both addition **and** concatenation. Any value that is input by a user using a prompt will always be a string. Therefore, JavaScript will regard num1 and num2 in this example as 2 strings, and therefore concatenate them. This is why you are seeing an incorrect result.

Therefore, if you wish to have these 2 inputs converted to numbers, you will need to “tell” JavaScript that these are numbers, and not strings. To do this, if you add the following code around your variable, then javascript will convert the variables to numbers:

To convert the variable called num1 to a number, you will need to express it as shown:

Number(num1)

There are other ways of converting strings to numbers. An example of some of these ways are:

```
+num1  
(num1 * 1)  
parseInt(num1)
```

Test your code to ensure that it is now working correctly.

2. Modify the above code so that it performs a range of calculations, including addition, subtraction, multiplication, division, modulus, increment and decrement – and outputs the results (correctly) to the webpage.

Exercise 6: Area of Rectangle

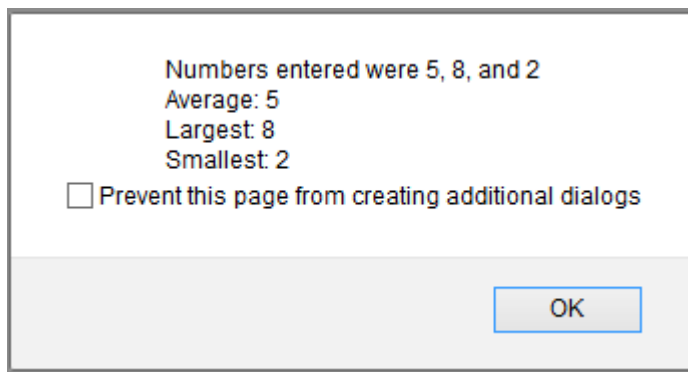
Create a script which calculates the area of a rectangle. Prompt the user for each of the following:

- Length
- Width
- Unit of measurement

Use an alert to display the area of the rectangle.

Exercise 7: Largest/Smallest Number, Average

Write a script that takes three integers from the user and displays the average, smallest and largest of the numbers in an alert message. An example output is as follows:



Exercise 8: Positive, Negative, Zero

Write a script that prompts a user to enter a number and determine if the number entered is a positive number, a negative number, or if it is equal to zero. Output the result of that determination to the screen.

Exercise 9: Random Numbers

The following code returns a number from 1 to 10:

```
Math.floor(Math.random() * 10) + 1; // returns a random integer from 1 to 10
```

Write a JavaScript program which uses this code to generate a random number. Implement an if-else statement to confirm the number returned by this code is greater than or equal to 1 and less than or equal to 10. In other words confirm this code does what you expect it to do. If it does display an alert message with the number, otherwise display a message in the alert which indicates the code did not do as expected (you should never see this message).

Note: To implement an if-else statement in JavaScript please refer to [if...else - JavaScript | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else).

Exercise 10: Guessing Game 1

Using the code from the previous exercise write a JavaScript program where the program generates a random integer between 1 to 10. The user is then prompted to input a guess number. If the user input matches with the random number generated, the program will display a message "Good Guess", otherwise it will display a message "Sorry, wrong number".

Exercise 11: Guessing Game 2

Modify the program from exercise 7 to allow multiple guesses. If the user input matches with guess number, the program will display a message "Good Guess" otherwise it will display a message "Try again" and will prompt the user to input another guess.

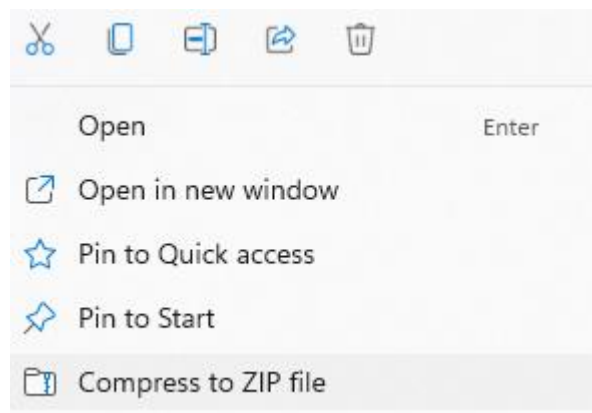
Note: For this solution you could implement a while loop – see [while - JavaScript | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/while).

Exercise 12: Guessing Game 3

Write a JavaScript Guessing Game program where the program takes a random integer between 1 to 100. The user is prompted to input a guess number. The user has six attempts to guess the right number. After each attempt, if the number is incorrect, the program will display "too high" or "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.

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.