JavaScript Functions

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.

Exercise 1:

In this exercise you are required to create functions and to run those functions to output values to a webpage. You can refer to the Functions examples from Lecture in Week 15 section on Moodle to help with this exercise – see the zip file.
In this exercise rather than embedding the JavaScript into the web page, use an external JavaScript file. To use an external JavaScript file, you just need to assign the name of the file, use a .js extension on the file, to the script elements src attribute. The src attribute specifies the url of an external script file.

For example:

```
<script src="Exercise1.js"></script>
```

Create a html page Exercise1.html, which calls a JavaScript file Exercise1.js which includes functions for the requirements below:

a. Create a function which prints out a message of text to the webpage (Use document.write() for this). Name the function, function1, and include code in this message to print out “Hello from function1” to the webpage.
b. Create a function which prints out a message of text to the webpage. Name the function, function2, and include code in this message to print out “Hello from function2” to the webpage.
c. Create a function, function3, which accepts one parameter, and prints out the value of that parameter to the webpage. For example, if you pass a value of 4 to the function, then it should print out “You passed me, function3, a value of: 4”.
d. Create a function, function4, which accepts two parameters, and prints out the values of those parameters to the webpage. For example, if you pass values of 10 and “Ten” to the function, then it should print out “You passed me, function4, values of: 10 and Ten”.
e. Create a function, function5, which accepts three parameters, and prints out the values of those parameters to the webpage. For example, if you pass values of 10, 20 and 30 to the function, then it should print out “You passed me, function5, values of: 10, 20 and 30”.

f. Create a function, function6, which accepts two parameters for first name and last name and prints out the values of those parameters to the webpage. For example, if you pass values of “Kevin” and “O’Brien” to the function, then it should print out “You passed me, function6, the name: Kevin O’Brien”.
g. Reuse function6. Prompt the user to enter a first name and last name. Pass the values entered by the user to function6.
h. Create a function, function7, which returns a number. Print out to the webpage that number returned by the function. For example, if function7 returns a value of 1000, then print out “function7 returned a value of 1000 to me”.
i. Create a function, function8, which accepts two parameters for first name and last name. Your function should return the full name. Print out the full name onto the webpage that is returned by this function. For example, if you pass the values “John” and “Smith” to function8, then the function should return the full name – “John Smith” and this should be printed to the webpage.
j. Create a function, function9, which accepts three parameters. Your function should return the sum of the three parameters. Print out the result onto the webpage that is returned by this function. For example, if you pass the values 4, 6, and 10 to function9, then print out the returned result – 20. Implement this function as an arrow function.

Exercise 2:

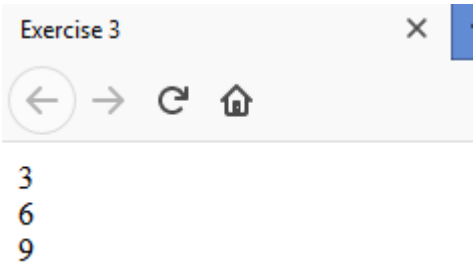
Write a JavaScript program to prompt a user to enter a numeric result for an exam. Using if-else condition, determine whether the result entered by the user is a merit, pass or fail grade. Each grade is determined as follows:

- Merit > 84
- Pass > 39
- Fail < 40

Output the grade to the browser window using HTML5 <h3> tags. Please ensure you use separate HTML and JavaScript files for this solution.

Exercise 3:

Write a JavaScript program which implements a loop that iterates 10 times. On each iteration determine whether the number which represents the current iteration is divisible by 3. If it is, then print that number to the webpage. See example output below:

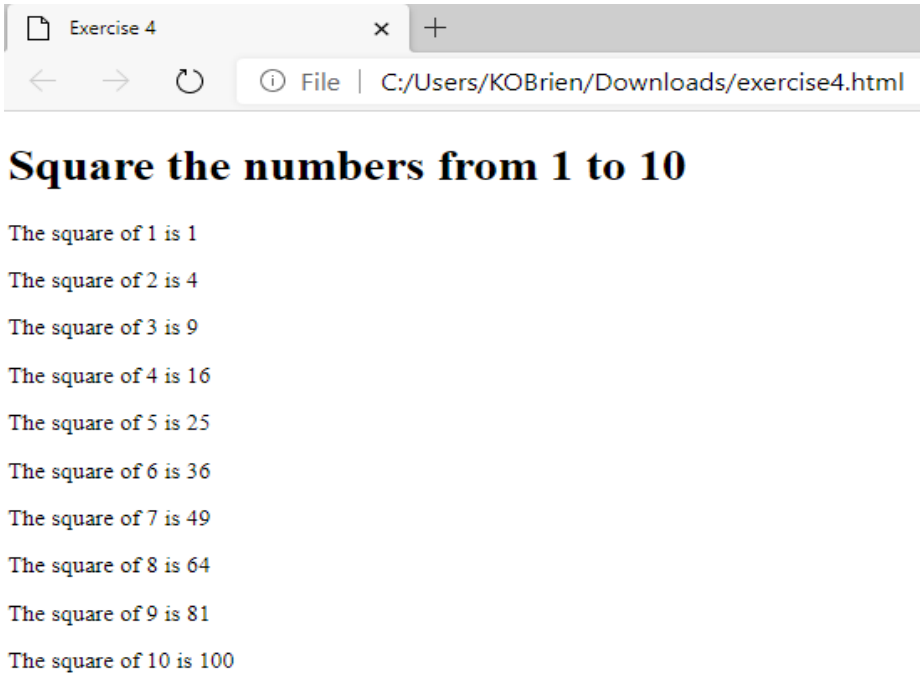


Please ensure you use separate HTML and JavaScript files for this solution.

Exercise 4:

Write a JavaScript function, square(number), which returns the result of calculating the square of the number which is passed to it. Use a for loop to call this function 10 times which passes all numbers from 1 to 10 to the function and displays the results of squaring each of these numbers in the browser.

An example output of this program is as follows:



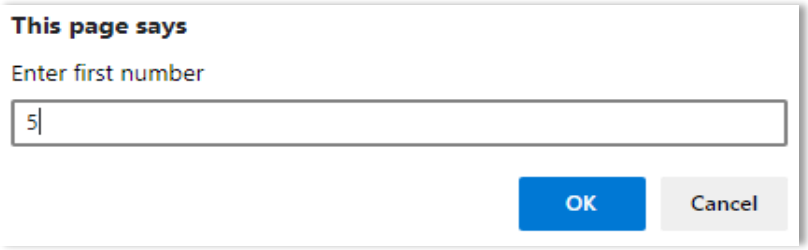
Please ensure you use separate HTML and JavaScript files for this solution.

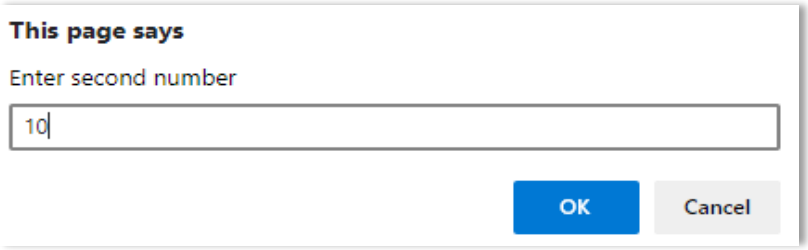
Exercise 5:

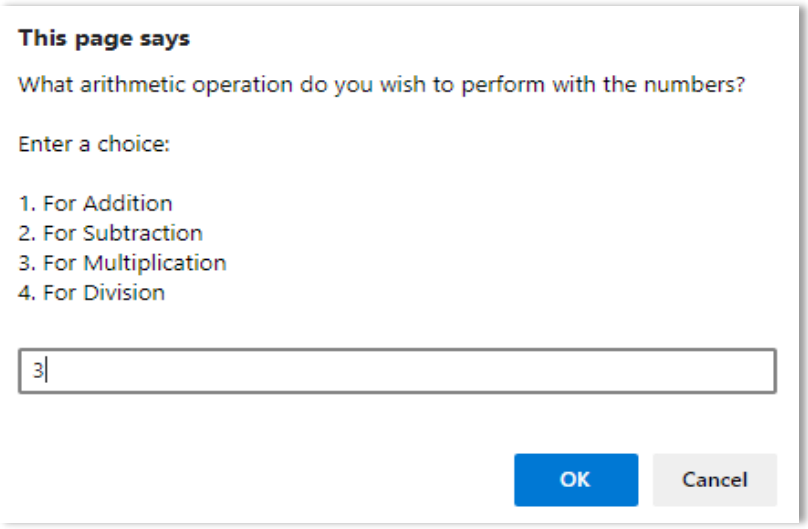
Implement a solution using JavaScript which does the following:

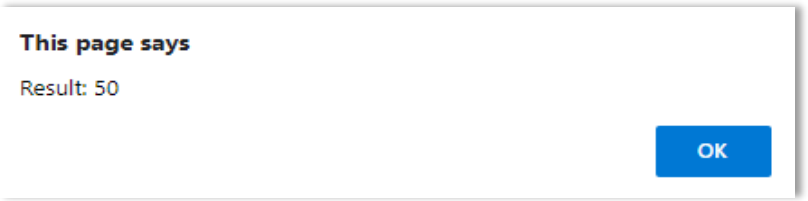
a. Prompt a user to enter two numbers.
b. Then prompt the user what arithmetic operation they wish to perform on the two numbers captured by the prompts in a. User should be prompted to enter any number from 1 to 4:
    1. For Addition
    2. For Subtraction
    3. For Multiplication
    4. For Division
c. Create a function named arithmetic, which has 3 parameters – 2 parameters for the numbers input by the user, and a third parameter for the number indicating arithmetic operation to perform. This function should include logic which determines what arithmetic operation to perform on the numbers input by the user in step a and return that result.
d. Use an alert to display the result returned by the function in step c.
e. Prompt the user, if they want to perform any more calculations, and if yes, repeat steps a to d. If not then the program ends. You will need to implement a loop in your solution for this.

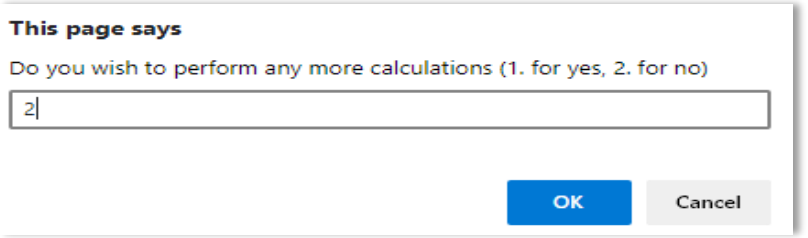
Example output from this program is as follows:









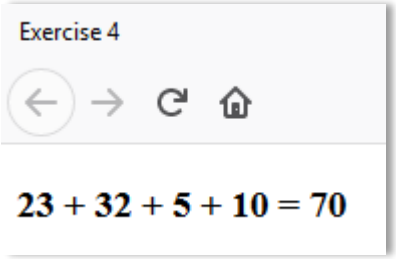


Please ensure you use separate HTML and JavaScript files for this solution.

Exercise 6:

Write a JavaScript function, getSum(number1, number2, number3, number4), which returns the result of adding four numbers together. Prompt the user for the four numbers and pass the numbers

entered by the user to the function. Display the result returned by the function on a web page. Output should be similar to the following:



Exercise 7:

For this exercise you are required to complete the following four tasks.

1. Create a JavaScript function named welcome, which contains code to display an alert with the following message:

   "Welcome to area calculator"

2. Create a JavaScript function area, which accepts two input parameters – one for length and one for width. The function should return the result of the product of these parameters – length * width.
3. Write JavaScript code to prompt a user to enter values for length and width – two prompts are required. The values entered should then be passed to the area function, and the result returned should be output to the web page.
4. Using the values entered by the user at the prompts draw out a rectangle shape on the screen. Use star(*) symbols for this. For example if the user entered a value of 4 for length, and a value of 5 for width, then print out 5 rows of stars symbols(*) with 4 star symbols(*) on each row. You could use a nested loop to achieve this output.

Exercise 8:

For this exercise you are required to complete the following four tasks.

1. Create a JavaScript function named greeting, which contains code to display the following message on a web page:

   "Welcome to the triangle"

2. Create a JavaScript function validator, which accepts one input parameter for a number. The function should check if the number passed to it when it is called is a valid number or not. The function should return this result – valid number or not a valid number. The rule to determine if it is a valid number is that the number must be any value from 2 to 9.
3. Write JavaScript code to prompt a user to enter a value for number of rows in a triangle. The value entered should then be passed to the validator function.
4. Using the value returned from the validator function, generate an output for the screen. The output will depend on the value returned from the function:
   a. If the value returned indicates the number was not a valid number entered by the user (was not a value from 2 to 9), then print a message to the screen indicating this.

The actual number entered by the user should be included in this message written to the screen.

b. If the value returned indicates the number was a valid number entered by the user (was a value from 2 to 9), then draw out a triangle shape on the screen. Use star(*) symbols for this. The triangle should contain as many rows as that entered by the user at the prompt. For example if the user entered 5, then the triangle should contain 5 rows. The first row, should contain 1 star symbol(*), the second row, should contain two star symbols(*) etc. all the way to the fifth row which should contain five star symbols(*).