JavaScript
Intro for Web Development
Lab 2

1 Conditional statements

1.1 if , if else, if else if

Files to use:
`if-else-if-basic.js`
`if-else-if-more.js`

Exercise: `if-else-if-question.js`
Answer: `if-else-if-answer.js`

JavaScript provide the if, if-else and if-else-if conditional statements to allow a computer to make decisions.

You can designate a block of statements delimited by a { } to be executed depending on the outcome of a conditional statement

The if-else-if is great for implementing logic for range classifications such as the one below:

BMI	CLASSIFICATION	HEALTH RISK
< 18.5	Underweight	Minimal
18.5 - 24.9	Normal	Minimal
25 - 29.9	Overweight	Increased
30 - 34.9	Obese	High
35 - 39.9	Severely Obese	Very High
> 40	Morbidly Obese	Extremely High

1.2 Ternary / conditional operator

File to use: `ternary-operator.js`

The ternary / conditional operator provides a short cut to implementing the logic of an if-else statement

1.3 switch case

File to use: `switch-basic.js`
Exercise: `switch-question.js`
Answer: `switch-answer.js`

The switch case provides a simpler way to implement the logic of a long if-elseif-else construct.

2 Loops

2.1 for loop

File to use: `for-loop.js`
Exercise: `for-loop-question.js`
Answer: `for-loop-answer.js`

The for loop is the most widely used iteration structure in most programming languages

2.2 while, do - while

File to use: `while-do-while.js`
Exercise: `while-do-while-question.js`
Exercise: `while-do-while-answer.js`

The while and do-while loop are used as an alternative to the for loop, but they are functionally equivalent

2.3 break and continue

Files to use:
`break-loop.js`
`continue-loop.js`

Exercise: `break-question.js`
Exercise: `break-answer.js`

The break keyword is used to immediately exit the loop body, while the continue keyword is used to jump to the end of the loop

3 Functions

3.1 Basics

File to use: `functions-basic.js`
Exercise: `functions-question.js`
Answer: `functions-answer.js`

A function is a self-contained group of program statements that accomplish a very specific task. This task could range from something very simple (for e.g. adding 2 numbers, finding the largest number in a list of 4 numbers, etc) to very complex.

By grouping all the code that accomplishes a specific task into a function, we can avoid replicating our code every time we need to perform that task. We can simply write a single statement that calls or invokes that particular function.

There are a variety of useful math related functions that are available directly in JavaScript to perform basic maths functions.

3.2 Anonymous functions and function expressions

File to use: `anonymous-function-expression.js`

An anonymous function is a function that has no name. The primary purpose of giving a function a name is so that we can use its name to invoke it repeatedly, thereby saving us the need to unnecessarily duplicate code. However, if the body of a specific function is very short (for e.g. only 1 or 2 statements) and that function is intended to be used only once, we can make it an anonymous function through the use of a function expression.

Functions are first class citizens in JavaScript. This means that you can store functions in variables, pass them to other functions as arguments, and return them from other functions as values. Typically when you pass functions as arguments to other functions, these functions are passed in the form of anonymous functions.

The JavaScript language comes with a lot of predefined functions (i.e. the functions were defined by the people who created that language for developers to use straight away). A commonly used predefined function in JavaScript is the setTimeoutfunction, which is used to set a count down timer. This function takes another function as its first parameter and the number of milliseconds to countdown as its second parameter. When the countdown completes, the parameter function is executed.

This parameter function is termed a callback . Callback functions help us to accomplish asynchronous programming in JavaScript applications that run in the browser, such as Angular or React

3.3 Arrow functions

File to use: `arrow-functions.js`
Exercise: `arrow-question.js`
Answer: `arrow-answer.js`

Arrow functions are a new language feature introduced in ES6 that provide you with an alternative way to write a shorter syntax compared to the function expression. Arrow function syntax is widely used throughout JavaScript applications, particular in applications developed using frameworks such as Angular or React