

Lab session 6 – JavaScript (part 1)

Unit	Programming languages: principles and design (6G6Z1110) Programming languages – SE frameworks (6G6Z1115)
Lecturer	Rob Frampton
Week	6
Portfolio element	This lab is not part of the portfolio.

Description

The aim of this lab exercise is to practise the JavaScript elements studied in the lecture “JavaScript – part 1”. By the end of this lab session, you should:

1. Be able to write and execute simple JavaScript programs
2. Be able to use flow control structures, arrays, functions and anonymous functions

Exercises

Exercise 1:

Write a program in JavaScript using a “for” statement that outputs the numbers between 1 and 200 but replaces any number divisible by three with the word “**Fizz**”, and any number divisible by five with the word “**Buzz**”. Words divisible by both three and five should output the word “**FizzBuzz**”.

Exercise 2:

Write a program which “flips a coin” in a loop. Inside a while loop, generate a random number using [Math.random\(\)](#). If the random number is greater than 0.5, display “Heads!” and keep looping. If the number is less than 0.5, display “Tails!” and stop looping. An example output would be:

Heads!
Heads!
Tails!

Exercise 3:

Write a function to compute the first n Fibonacci numbers, returning them in an array. The pseudocode for computing the first n Fibonacci numbers looks like this:

```
function computeFibonacci( $n$ )  
     $array := [ 0, 1 ]$   
  
    for  $i := 2$  to  $n$   
         $nextNumber = array[i - 1] + array[i - 2]$   
        Append  $nextNumber$  to  $array$   
    end  
end
```

Call your function to compute the first 20 numbers, and display the output to the console. It should look like this:

```
[ 0,
  1,
  1,
  2,
  3,
  5,
  8,
  13,
  21,
  34,
  55,
  89,
  144,
  233,
  377,
  610,
  987,
  1597,
  2584,
  4181 ]
```

Exercise 4:

Write a function called `twoTimesTable` which takes an input, `x`, and displays the following text in the console:

```
Two times x is y
```

where `x` and `y` are replaced with the appropriate values.

Now write a function called `iterate` which takes two arguments: `action` and `count`. The `iterate` function assumes that `count` is an integer and that `action` is a function. Write a `for` loop that iterates `count` times, calling the `action` function each iteration, passing in the number of iterations so far (i.e. the loop counter) as an argument.

Finally, your program should call the `iterate` function, passing in the `twoTimesTable` function as an argument like this:

```
iterate(twoTimesTable, 10)
```

which should result in the output:

```
Two times 0 is 0
Two times 1 is 2
Two times 2 is 4
Two times 3 is 6
Two times 4 is 8
Two times 5 is 10
Two times 6 is 12
Two times 7 is 14
Two times 8 is 16
Two times 9 is 18
```

Exercise 5a:

Write a JavaScript program which creates an array and initializes it with the strings “Cat”, “Dog”, “Cow” and “Rabb i t”. Iterate over the array using a `for` loop, displaying each word to the console, except the word should be in upper case.

Hint: you will need the [toUpperCase\(\)](#) function on the [String](#) type

Exercise 5b:

Modify the program from Exercise 5a to use the [forEach\(\)](#) function on the [Array](#) type instead of a `for` loop. You should pass it an anonymous function using the `function` notation.

Exercise 5c:

Modify the program from Exercise 5b to use the arrow notation for anonymous functions instead of the `function` notation. You should be able to write this program without using any curly braces (`{ }`).

Extension Exercise (Optional):

Use the popular [express](#) framework to build a web server in node. In a new directory, execute the following commands: