# Javascript Level 1 Exercises
# Set 1

## #1. FizzBuzz

Write a JavaScript program which iterates the integers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

## #2. If you can't sleep, just count sheeps!!

Given a number, 3 for example, return a string with a murmur: "1 sheep...2 sheep...3 sheep..."

Example:

```
INPUT: countSheep(3)
OUTPUT: "1 sheep...2 sheep...3 sheep…"

INPUT: countSheep(5)
OUTPUT: "1 sheep...2 sheep...3 sheep...4 sheep...5 sheep..."
```

Note:
You will always receive a positive integer.

## #3. Count Up then Down

Create a function that takes a number as an argument and returns an array. The first element of the array should be 0, and then each element should increase by 1 until it reaches the input number. Then, each element should count back down to 0.

Examples:

```
INPUT: countUpThenDown(2)
OUTPUT: [0, 1, 2, 1, 0]

INPUT: countUpThenDown(0)
OUTPUT: [0]
```

## #4. Find the sum of the array

Given an array of numbers, find the sum of the elements

Example:

```
INPUT: sumArray([10,20,30])
OUTPUT: 60

INPUT: sumArray([])
OUTPUT: 0
```

## #5. Count the Number of Vowels

Write a JavaScript function that accepts a string as a parameter and counts the number of vowels within the string.

Example:

```
INPUT: countVowels('The quick brown fox')
OUTPUT: 5

INPUT: countVowels('javascript is awesome')
OUTPUT: 8
```

Note : As the letter 'y' can be regarded as both a vowel and a consonant, we do not count 'y' as vowel here.

## #6. Generate an Array by Range

Write a JavaScript function to generate an array between two integers of 1 step length.

Example:

```
INPUT: rangeBetween(4, 7)
OUTPUT: [4, 5, 6, 7]

INPUT: rangeBetween(-4, 7)
OUTPUT: [-4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7]
```

## #7. Chop a string

Write a JavaScript function to chop a string into chunks of a given length.

Example:

```
INPUT: stringChop('resource',2)
OUTPUT:  [ "re", "so", "ur", "ce"]

INPUT: stringChop('dcresource',3)
OUTPUT: ["dcr", "eso", "urc", "e"]
```

## #8. First n elements in array

Write a JavaScript function to get the first element of an array. Passing a parameter 'n' will return the first 'n' elements of the array.

Example:

```
INPUT: first([7, 9, 0, -2])
OUTPUT: 7

INPUT: first([],3)
OUTPUT: []

INPUT: first([7, 9, 0, -2],3)
OUTPUT: [7, 9, 0]

INPUT: first([7, 9, 0, -2],6)
OUTPUT: [7, 9, 0, -2]

INPUT: first([7, 9, 0, -2],-3)
OUTPUT: [7,9,0,-2]
```