**Heading**: Convert Celsius to Fahrenheit

**Description**: The algorithm to convert from Celsius to Fahrenheit is the temperature in Celsius times 9/5, plus 32.

You are given a variable celsius representing a temperature in Celsius. Use the variable fahrenheit already defined and assign it the Fahrenheit temperature equivalent to the given Celsius temperature. Use the algorithm mentioned above to help convert the Celsius temperature to Fahrenheit.

Don't worry too much about the function and return statements as they will be covered in future challenges. For now, only use operators that you have already learned.

**Hint**: You are given a variable **celsius** representing a temperature in Celsius. Use the variable **fahrenheit** already defined and apply the algorithm to assign it the corresponding temperature in Fahrenheit.

function convertToF(celsius) {

 let fahrenheit ;

 return fahrenheit;

}

convertToF(20)should return a value of 68

**Heading**: Factorialize a Number

**Description**: Return the factorial of the provided integer.

If the integer is represented with the letter n, a factorial is the product of all positive integers less than or equal to n.

Factorials are often represented with the shorthand notation n!

For example: 5! = 1 \* 2 \* 3 \* 4 \* 5 = 120

Only integers greater than or equal to zero will be supplied to the function.

**Hint**: Return the factorial of the provided integer. If the integer is represented with the letter n, a factorial is the product of all positive integers less than or equal to n.

Factorials are often represented with the shorthand notation n!

For example: 5! = 1 \* 2 \* 3 \* 4 \* 5 = 120

**Hint: 1**

This one starts easily since 0! = 1, so you can go ahead and simply return 1 there.

We can use that as an if in order to break the loop we’re going to create using a **recursive function**. It will check if the number you gave the function is 0 (which would be the end of your factorial chain). Functions “end” when they return anything. In fact, **all** functions without an explicit return statement will return undefined.

**Heading**: Return Largest Numbers in Arrays

**Description**: Return an array consisting of the largest number from each provided sub-array. For simplicity, the provided array will contain exactly 4 sub-arrays.

Remember, you can iterate through an array with a simple for loop, and access each member with array syntax arr[i].

**Hint**: You will get an array that contains sub arrays of numbers and you need to return an array with the largest number from each of the sub arrays.

**Hint: 1**

You will need to keep track of the array with the answer and the largest number of each sub-array.

Ex => var arr = [ [3,4,5],  [5,6,7], [4,5,6], [45,65,78] ]

Result => [5,7,6,78]

**Heading**: Repeat a String Repeat a String

**Description**: Repeat a given string str(first argument) for numtimes (second argument). Return an empty string if numis not a positive number.

**Hint**: The program is very simple, we have to take a variable and return that variable being repeated certain amount of times. No need to add space or anything, just keep repeating it into one single string.

**Hint: 1**

You can’t edit strings, you will need to create a variable to store the new string.

Ex: repeatStringNumTimes("abc", 3)should return "abcabcabc"

**Heading**: Boo who

**Description**: Write a function to check if a value is classified as a boolean primitive. Return true or false.

Boolean primitives are true and false.

**Hint**: You will need to check for the type of the parameter to see if it is a boolean.

Ex: booWho(false)should return true.

      booWho([1, 2, 3])should return false.

**Heading**: Title Case a Sentence

**Description**: Return the provided string with the first letter of each word capitalized. Make sure the rest of the word is in lowercase.

For the purpose of this exercise, you should also capitalize connecting words like "the" and "of".

**Hint**: You should start by splitting the string into an array of words.

Split the sentence.

Ex: frankenSplice([1, 2, 3], [4, 5], 1) should return  [4, 1, 2, 3, 5]

**Heading**: Falsy Bouncer

**Description**: Remove all falsy values from an array.

Falsy values in JavaScript are false, null, 0, "", undefined, and NaN.

Hint: Try converting each value to a Boolean.

**Hint**: Falsy is something which evaluates to FALSE. There are only six falsy values in JavaScript: undefined, null, NaN, 0, "" (empty string), and false of course.

Ex: bouncer([7, "ate", "", false, 9])should return [7, "ate", 9].