

## L. Functions:

1. Write a function called `'describeCountry'` which takes three parameters: `'country'`, `'population'` and `'capitalCity'`. Based on this input, the function returns a string with this format: `'Finland has 6 million people and its capital city is Helsinki'`.
2. Call this function 3 times, with input data for 3 different countries. Store the returned values in 3 different variables, and log them to the console.

## M. General Functions vs. Expressions

1. The world population is 7900 million people. Create a **function declaration** called `'percentageOfWorld1'` which receives a `'population'` value, and returns the percentage of the world population that the given population represents. For example, China has 1441 million people, so it's about 18.2% of the world's population.
2. To calculate the percentage, divide the given `'population'` value by 7900 and then multiply by 100.
3. Call `'percentageOfWorld1'` for 3 populations of countries of your choice, store the results into variables, and log them to the console.
4. Create a **function expression** that does the exact same thing, called `'percentageOfWorld2'`, and also call it with 3 country populations (can be the same populations).

## N. Arrow Functions

1. Recreate the last assignment, but this time create an arrow function called `'percentageOfWorld3'`

## O. Functions Calling Other Functions

1. Create a function called `'describePopulation'`. Use the function type you like the most. This function takes in two arguments: `'country'` and `'population'`, and returns a string like this: `'China has 1441 million people, which is about 18.2% of the world.'`
2. To calculate the percentage, `'describePopulation'` call the `'percentageOfWorld1'` you created earlier.
3. Call `'describePopulation'` with data for 3 countries of your choice.

## P. Introduction to Arrays

1. Create an array containing 4 population values of 4 countries of your choice. You may use the values you have been using previously. Store this array into a variable called `'populations'`.
2. Log to the console whether the array has 4 elements or not (true or false).
3. Create an array called `'percentages'` containing the percentages of the world population for these 4 population values. Use the function `'percentageOfWorld'` that you created earlier to compute the 4 percentage values.

## Q. Basic Array Operations (Methods)

1. Create an array containing all the neighbouring countries of a country of your choice. Choose a country which has at least 2 or 3 neighbours. Store the array into a variable called `'neighbours'`.
2. At some point, a new country called `'Utopia'` is created in the neighbourhood of your selected country. So add it to the end of the `'neighbours'` array.
3. Unfortunately, after some time, the new country is dissolved. So remove it from the end of the array.
4. If the `'neighbours'` array does not include the country `'Germany'`, log to the console: `'Probably not a central European country.'`
5. Change the name of one of your neighbouring countries. To do that, find the index of the country in the `'neighbours'` array, and then use that index to change the array at that index position. For example, you can search for `'Sweden'` in the array, and then replace it with `'Republic of Sweden'`.

## R. Introduction to Objects

1. Create an object called `'myCountry'` for a country of your choice, containing properties `'country'`, `'capital'`, `'language'`, `'population'` and `'neighbours'` (an array like we used in previous assignments).

## S. Dot vs. Bracket Notation

1. Using the object from the previous assignment, log a string like this to the console: 'Finland has 6 million finnish-speaking people, 3 neighbouring countries and a capital called Helsinki.'
2. Increase the country's population by two million using dot notation, and then decrease it by two million using brackets notation.

## T. Object Methods

1. Add a method called 'describe' to the 'myCountry' object. This method will log a string to the console, similar to the string logged in the previous assignment, but this time using the 'this' keyword.
2. Call the 'describe' method
3. Add a method called 'checkIsland' to the 'myCountry' object. This method will set a new property on the object, called 'island'. 'island' will be true if there are no neighbouring countries, and false if there are. Use the ternary operator to set the property.

## U. Iteration: The for Loop

1. There are elections in your country! In a small town, there are only 50 voters. Use a for loop to simulate the 50 people voting, by logging a string like this to the console (for numbers 1 to 50): 'Voter number 1 is currently voting'

## V. Looping Arrays, Breaking and Continuing

1. Let's bring back the 'populations' array from a previous assignment
2. Use a for loop to compute an array called 'percentages2' containing the percentages of the world population for the 4 population values. Use the function 'percentageOfWorld1' that you created earlier
3. Confirm that 'percentages2' contains exactly the same values as the 'percentages' array that we created manually in the previous assignment, and reflect on how much better this solution is.

## W. Looping Backwards and Loops in Loops

1. Store this array of arrays into a variable called 'listOfNeighbours'

`[['Canada', 'Mexico'], ['Spain'], ['Norway', 'Sweden', 'Russia']];`

2. Log only the neighbouring countries to the console, one by one, not the entire arrays. Log a string like 'Neighbour: Canada' for each country

3. You will need a loop inside a loop for this. This is actually a bit tricky, so don't worry if it's too difficult for you! But you can still try to figure this out anyway 😊

## X. The while Loop

1. Recreate the challenge from the lecture 'Looping Arrays, Breaking and Continuing', but this time using a while loop (call the array 'percentages3')

2. Reflect on what solution you like better for this task: the for loop or the while loop?