

Solutions – Part 1

LECTURE: Values and Variables

```
let country = 'Portugal';  
let continent = 'Europe';  
let population = 10;  
console.log(country);  
console.log(continent);  
console.log(population);
```

LECTURE: Data Types

```
let isIsland = false;  
let language;  
console.log(typeof isIsland);  
console.log(typeof population);  
console.log(typeof country);  
console.log(typeof language);
```

LECTURE: let, const and var

```
language = 'portuguese';  
const country = 'Portugal';  
const continent = 'Europe';  
const isIsland = false;  
isIsland = true;
```

LECTURE: Basic Operators

```
console.log(population / 2);
population++;
console.log(population);
console.log(population > 6);
console.log(population < 33);
const description1 =
  country +
  ' is in ' +
  continent +
  ', and its ' +
  population +
  ' million people speak ' +
  language;
console.log(description1);
```

LECTURE: Strings and Template Literals

```
const description = `${country} is in ${continent}, and its
${population} million people speak ${language}`;
```

LECTURE: Taking Decisions: if / else Statements

```
if (population > 33) {
  console.log(`${country}'s population is above average`);
} else {
  console.log(
    `${country}'s population is ${33 - population} million
    below average`,
  );
}
```

LECTURE: Type Conversion and Coercion

```
console.log('9' - '5'); // -> 4
console.log('19' - '13' + '17'); // -> '617'
console.log('19' - '13' + 17); // -> 23
console.log('123' < 57); // -> false
console.log(5 + 6 + '4' + 9 - 4 - 2); // -> 1143
```

LECTURE: Equality Operators: == vs. ===

```
const numNeighbours = prompt(
  'How many neighbour countries does your country have?',
);

// LATER : This helps us prevent bugs
const numNeighbours = Number(
  prompt('How many neighbour countries does your country
  have?'),
);

if (numNeighbours === 1) {
  console.log('Only 1 border!');
} else if (numNeighbours > 1) {
  console.log('More than 1 border');
} else {
  console.log('No borders');
}
```

LECTURE: Logical Operators

```
if (language === 'english' && population < 50 && !isIsland)
{
  console.log(`You should live in ${country} :)`);
} else {
  console.log(`${country} does not meet your criteria :(`);
}
```

LECTURE: The switch Statement

```
switch (language) {
  case 'chinese':
  case 'mandarin':
    console.log('MOST number of native speakers!');
    break;
  case 'spanish':
    console.log('2nd place in number of native speakers');
    break;
  case 'english':
    console.log('3rd place');
    break;
  case 'hindi':
    console.log('Number 4');
    break;
  case 'arabic':
    console.log('5th most spoken language');
    break;
  default:
    console.log('Great language too :D');
}
```

LECTURE: The Conditional (Ternary) Operator

```
console.log(  
  `${country}'s population is ${population > 33 ? 'above' :  
    'below'} average`,  
);
```

JavaScript Fundamentals – Part 2

Note: Please start Part 2 from scratch and comment out all the code from Part 1.

LECTURE: 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

LECTURE: Function Declarations 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 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** which does the exact same thing, called `'percentageOfWorld2'`, and also call it with 3 country populations (can be the same populations)

LECTURE: Arrow Functions

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

LECTURE: 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

LECTURE: 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 'percentageOfWorld1' that you created earlier to compute the 4 percentage values

LECTURE: 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 :D'*
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'.

LECTURE: 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)

LECTURE: 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**.

LECTURE: 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 'isIsland'. 'isIsland' will be true if there are no neighbouring countries, and false if there are. Use the ternary operator to set the property.

LECTURE: 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'*

LECTURE: 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

LECTURE: 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 😊

LECTURE: 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?

Solutions – Part 2

LECTURE: Functions

```
function describeCountry(country, population, capitalCity) {  
  return `${country} has ${population} million people and  
    its capital city is ${capitalCity}`;  
}  
  
const descPortugal = describeCountry('Portugal', 10,  
  'Lisbon');  
const descGermany = describeCountry('Germany', 83,  
  'Berlin');  
const descFinland = describeCountry('Finland', 6,  
  'Helsinki');  
console.log(descPortugal, descGermany, descFinland);
```

LECTURE: Function Declarations vs. Expressions

```
function percentageOfWorld1(population) {  
  return (population / 7900) * 100;  
}  
  
const percentageOfWorld2 = function (population) {  
  return (population / 7900) * 100;  
};  
  
const percPortugal1 = percentageOfWorld1(10);  
const percChina1 = percentageOfWorld1(1441);  
const percUSA1 = percentageOfWorld1(332);  
console.log(percPortugal1, percChina1, percUSA1);
```

LECTURE: Arrow Functions

```
const percentageOfWorld3 = population => (population / 7900)
* 100;

const percPortugal3 = percentageOfWorld3(10);
const percChina3 = percentageOfWorld3(1441);
const percUSA3 = percentageOfWorld3(332);
console.log(percPortugal3, percChina3, percUSA3);
```

LECTURE: Functions Calling Other Functions

```
const describePopulation = function (country, population) {
  const percentage = percentageOfWorld1(population);
  const description = `${country} has ${population} million
  people, which is about ${percentage}% of the world.`;
  console.log(description);
};

describePopulation('Portugal', 10);
describePopulation('China', 1441);
describePopulation('USA', 332);
```

LECTURE: Introduction to Arrays

```
const populations = [10, 1441, 332, 83];
console.log(populations.length === 4);
const percentages = [
  percentageOfWorld1(populations[0]),
  percentageOfWorld1(populations[1]),
  percentageOfWorld1(populations[2]),
  percentageOfWorld1(populations[3])
];
console.log(percentages);
```

LECTURE: Basic Array Operations (Methods)

```
const neighbours = ['Norway', 'Sweden', 'Russia'];

neighbours.push('Utopia');
console.log(neighbours);

neighbours.pop();
console.log(neighbours);

if (!neighbours.includes('Germany')) {
  console.log('Probably not a central European country :D');
}

neighbours[neighbours.indexOf('Sweden')] = 'Republic of Sweden';
console.log(neighbours);
```

LECTURE: Introduction to Objects

```
const myCountry = {  
  country: 'Finland',  
  capital: 'Helsinki',  
  language: 'finnish',  
  population: 6,  
  neighbours: ['Norway', 'Sweden', 'Russia']  
};
```

LECTURE: Dot vs. Bracket Notation

```
console.log(  
  `${myCountry.country} has ${myCountry.population} million  
  ${myCountry.language}-speaking people,  
  ${myCountry.neighbours.length} neighbouring countries and  
  a capital called ${myCountry.capital}.`  
);  
  
myCountry.population += 2;  
console.log(myCountry.population);  
  
myCountry['population'] -= 2;  
console.log(myCountry.population);
```

LECTURE: Object Methods

```
const myCountry = {
  country: 'Finland',
  capital: 'Helsinki',
  language: 'finnish',
  population: 6,
  neighbours: ['Norway', 'Sweden', 'Russia'],

  describe: function () {
    console.log(
      `${this.country} has ${this.population} million
      ${this.language}-speaking people,
      ${this.neighbours.length} neighbouring countries and a
      capital called ${this.capital}.`
    );
  },

  checkIsland: function () {
    this.isIsland = this.neighbours.length === 0 ? true :
    false;

    // Even simpler version (see why this works...)
    // this.isIsland = !Boolean(this.neighbours.length);
  }
};

myCountry2.describe();
myCountry2.checkIsland();
console.log(myCountry2);
```

LECTURE: Iteration: The for Loop

```
for (let voter = 1; voter <= 50; voter++)
  console.log(`Voter number ${voter} is currently voting`);
```

LECTURE: Looping Arrays, Breaking and Continuing

```
const populations = [10, 1441, 332, 83];
const percentages2 = [];
for (let i = 0; i < populations.length; i++) {
  const perc = percentageOfWorld1(populations[i]);
  percentages2.push(perc);
}
console.log(percentages2);
```

LECTURE: Looping Backwards and Loops in Loops

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

for (let i = 0; i < listOfNeighbours.length; i++)
  for (let y = 0; y < listOfNeighbours[i].length; y++)
    console.log(`Neighbour: ${listOfNeighbours[i][y]}`);
```

LECTURE: The while Loop

```
const percentages3 = [];
let i = 0;
while (i < populations.length) {
  const perc = percentageOfWorld1(populations[i]);
  percentages3.push(perc);
  i++;
}
console.log(percentages3);
```