JONAS.IO
SCHMEDTMANN

PRACTICE ASSIGNMENTS FOR JAVASCRIPT FUNDAMENTALS SECTIONS

THE COMPLETE JAVASCRIPT COURSE

@JONASSCHMEDTMAN

Table of Contents

Instructions

- There is **one assignment for each lecture** in the JavaScript Fundamentals Sections Part 1 and 2 (not all lectures, but most);
- The goal of these assignments is that you can immediately apply the concepts you learn in each video;
- So after you complete each lecture, find the assignment for the video you just watched, and write the code according to the instructions;
- Take all the time that you need, **no need to hurry**!
- The **solution** for each assignment is at the end of Part 1 and Part 2. I advise you to **check it out after you completed each assignment**, or in case you have trouble moving forward in the code;
- In order to actually write the code, **create a new script** called `assignments.js` in the current project folder and link it to the HTML file we have been using, just like we previously linked `script.js` (an HTML file can include multiple JavaScript scripts). The console will now show outputs from both `script.js` and `assignments.js` 😉
- And now, go have fun with these assignments! By the way, all these assignments are about countries 🇵🇹 🇩🇪 🇮🇳

JavaScript Fundamentals – Part 1

LECTURE: Values and Variables

1. Declare variables called `'country'`, `'continent'` and `'population'` and assign their values according to your own country (population in millions)
2. Log their values to the console

LECTURE: Data Types

1. Declare a variable called `'isIsland'` and set its value according to your country. The variable should hold a Boolean value. Also declare a variable `'language'`, but don't assign it any value yet
2. Log the types of `'isIsland'`, `'population'`, `'country'` and `'language'` to the console

LECTURE: let, const and var

1. Set the value of `'language'` to the language spoken where you live (some countries have multiple languages, but just choose one)
2. Think about which variables should be `const` variables (which values will never change, and which might change?). Then, change these variables to `const`.
3. Try to change one of the changed variables now, and observe what happens

LECTURE: Basic Operators

1. If your country split in half, and each half would contain half the population, then how many people would live in each half?
2. Increase the population of your country by 1 and log the result to the console
3. Finland has a population of 6 million. Does your country have more people than Finland?
4. The average population of a country is 33 million people. Does your country have less people than the average country?
5. Based on the variables you created, create a new variable `'description'` which contains a string with this format: *'Portugal is in Europe, and its 11 million people speak portuguese'*

LECTURE: Strings and Template Literals

1. Recreate the `'description'` variable from the last assignment, this time using the template literal syntax

LECTURE: Taking Decisions: if / else Statements

1. If your country's population is greater that 33 million, log a string like this to the console: *'Portugal's population is above average'*. Otherwise, log a string like *'Portugal's population is 22 million below average'* (the 22 is the average of 33 minus the country's population)
2. After checking the result, change the population temporarily to 13 and then to 130. See the different results, and set the population back to original

LECTURE: Type Conversion and Coercion

1. Predict the result of these 5 operations without executing them:
```
'9' - '5';
'19' - '13' + '17';
'19' - '13' + 17;
'123' < 57;
5 + 6 + '4' + 9 - 4 - 2;
```
2. Execute the operations to check if you were right

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

1. Declare a variable `'numNeighbours'` based on a prompt input like this: `prompt('How many neighbour countries does your country have?');`
2. If there is only 1 neighbour, log to the console *'Only 1 border!'* (use loose equality == for now)
3. Use an `else-if` block to log 'More than 1 border' in case `'numNeighbours'` is greater than 1
4. Use an `else` block to log *'No borders'* (this block will be executed when `'numNeighbours'` is 0 or any other value)
5. Test the code with different values of `'numNeighbours'`, including 1 and 0.
6. Change == to ===, and test the code again, with the same values of `'numNeighbours'`. Notice what happens when there is exactly 1 border! Why is this happening?
7. Finally, convert `'numNeighbours'` to a number, and watch what happens now when you input 1
8. Reflect on why we should use the === operator and type conversion in this situation

LECTURE: Logical Operators

1. Comment out the previous code so the prompt doesn't get in the way
2. Let's say Sarah is looking for a new country to live in. She wants to live in a country that speaks english, has less than 50 million people and is not an island.
3. Write an `if` statement to help Sarah figure out if your country is right for her. You will need to write a condition that accounts for all of Sarah's criteria. Take your time with this, and check part of the solution if necessary.
4. If yours is the right country, log a string like this: *'You should live in Portugal :)'*. If not, log *'Portugal does not meet your criteria :('*
5. Probably your country does not meet all the criteria. So go back and temporarily change some variables in order to make the condition true (unless you live in Canada :D)

LECTURE: The switch Statement

1. Use a `switch` statement to log the following string for the given `'language'`:
   chinese or mandarin: *'MOST number of native speakers!'*
   spanish: *'2nd place in number of native speakers'*
   english: *'3rd place'*
   hindi: *'Number 4'*
   arabic: *'5th most spoken language'*
   for all other simply log *'Great language too :D'*

LECTURE: The Conditional (Ternary) Operator

1. If your country's population is greater than 33 million, use the ternary operator to log a string like this to the console: *'Portugal's population is above average'*. Otherwise, simply log *'Portugal's population is below average'*. Notice how only one word changes between these two sentences!
2. After checking the result, change the population temporarily to 13 and then to 130. See the different results, and set the population back to original

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);
```