

# Exercises



*MVT21*

JavaScript

## Goals

- Get familiar with modern JavaScript
- Be able to debug code using a debugger
- Be able to set up and implement unit tests using JEST
- Learn to set up a backend system using Node.js
- Get familiar with SQL and how to setup a database

# Contents

<b>1</b>	<b>Front-end JavaScript</b>	<b>2</b>
<b>2</b>	<b>Unit Tests</b>	<b>7</b>

# Chapter 1

## Front-end JavaScript

- 1.1 Your friend from abroad comes to visit, and she does not understand how to read a temperature scale in Celsius. Albeit being a novice programmer, you have the skills to implement a simple website converting Fahrenheit to Celsius, and vice versa. An example of this awesome project is shown below.

**My leet temp converter**

**result:**

**result:**

If you don't remember the formula for conversion, we give it here

$$T_C = \frac{T_F - 32}{1.8}$$

$$T_F = T_C \times 1.8 + 32$$

where  $T_C$  is the temperature in Celsius, and  $T_F$  is the temperature in Fahrenheit.

- 1.2 In a school project, you are to measure the temperature in your computer during the day, and calculate the average. You want to be able to input all values separated with a comma in a text field directly (not one at a time). Example below.

Enter all measured vaules you have taken, separated with comma, ","

**Result:**

**Hint:** Check out the `split()` function for strings, [https://www.w3schools.com/jsref/jsref\\_split.asp](https://www.w3schools.com/jsref/jsref_split.asp)

- 1.3 Your dusty friend Pythagoras comes to visit during his trip around Europe. Being a somewhat "old school" mathematician, he is not familiar with the modern number system. You decide to make it easier for him to order a cheese burger in a drive-through by programming a Roman  $\rightarrow$  Arabic converter (you do know that we are using Arabic numbers usually?). Example below.

Enter a Roman number to convert to Arabic

**Result:**

**Note** that this is quite a tricky challenge. To make it easier for you, only handle Roman numbers where you go from a larger number to a smaller, e.g., XVI, and not e.g., XIV as this is much harder.

**Hint:** You can of course create a bunch of if statements to convert all values, but try to use a map instead, example:

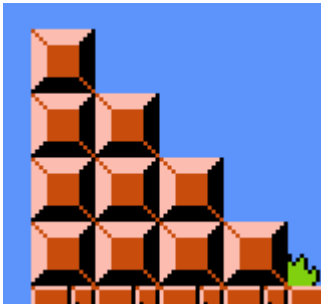
---

```
let roman = {"M": 1000, "D": 500, ...};
console.log(roman["D"]); // will print 500
// we can also do:
console.log(roman.D);
```

---

**Listing 1.1:** Example of using maps.

- 1.4 You are a fed up with all these modern games like Fartnite, Battleshit 2042, and Cyberprank. Since you have been taking a programming course, you decide to bring the old favourite, Super Mario Bros, back into life. You feel like designing the levels is a good place to start. The first objective is to implement the infamous obstacle, the stair.



Although it is possible to work with images in JavaScript, you use hashtag "#" as a block instead. Create a simple web page where you can enter the height of the stair, and print the result in the console. Example below.

**Note** that you have to use nested loops to solve this. <https://sebastian.com/nested-loops-javascript/>

draw

>\_ Console (beta) 5 0 0 0 0

"#"

"##"

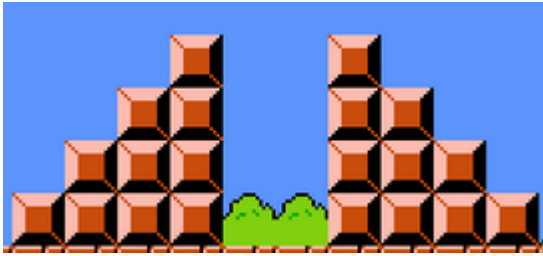
"###"

"####"

"#####"

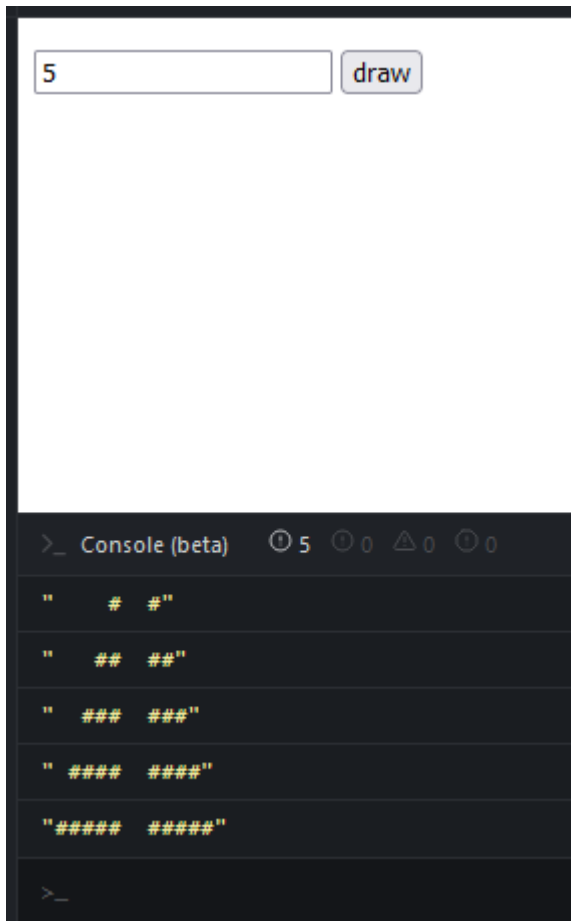
>\_

1.5 One stair is cool and all, but what is even cooler is the double stair.



This is similar to the previous exercise, but includes the double stair.

**Hint:** figure out how many spaces you need before the first "#" for the left stair. The right stair is the same code as the previous exercise.



1.6 The company Haribo is looking to launch a new family of sweets. Since their current best selling product MAOAM happens to be a palindrome, they believe that the next product name should also be a palindrome to be a top selling product.

Haribo knows that you are somewhat of an array expert in JavaScript, so they hire you to help with the product name. Your task is go implement a function that checks whether a string is a palindrome or not.

Make sure that both words and sentences work, such as:

- Madam
- Taco cat
- Yo, banana boy

It is common not to consider spaces, and punctuation as part of the sentence. So it would be a good idea to remove those characters first.

**Task:**

- Use an array to store the letters (not punctuation etc.)
  - Use the `reverse()` function on the array
- 1.7 Modify (or create a new version) the program from the previous example. This time, instead of using the `reverse()` function, utilize the stack functions, i.e., `push()` and `pop()`. Remember that when popping an element from a stack (array in this case) you get the latest added value.
- 1.8 You get hired at a startup company working with AI (Artificial Intelligence) for scanning barcodes. Your task is to detect if there are *at least* 3 consecutive identical values (3 in a row) in an image.

Another team handles the image analysis, and you are handed the input as a binary array, consisting of 1s (ones) and 0s (zeros). An example input is shown below.

[1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0]

[1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0]

In both of the cases, we should return true.

**Hint:** Use the `slice()` method to extract a sub-array of size 3. If all values are equal in this smaller array, you know that we have found our "three in a row" and may return true. See the section "Slicing an Array" here [https://www.w3schools.com/js/js\\_array\\_methods.asp](https://www.w3schools.com/js/js_array_methods.asp).

## Chapter 2

## Unit Tests