

JavaScript & Node.js Essentials for Test Automation

Cope Automation

Exercises

Exercise - 1

Declare three variables (e.g. day, month, year) in a single line and assign different values ("Monday", "Jan", 2023) and print them in the console

Exercise - 2

Assume a scenario where you wanted to check ***who is using what operating system*** (e.g. macOS or windows or others). Now, define a function (e.g. checkOS) with two parameters (e.g. personName and os) and perform the followings

1. Check if the user always provides two values (e.g personName as 'Rob' and os as 'Windows'. if not, throw error saying "One or more values are missing..."
2. If user provides other data type than string, then throw error message saying "Only string values are accepted, but received values are <val1> and <val2>".
3. Write a console message saying "Hello <name>, you have provided <os> as your operating system..."
4. If user provides any of the below values as os, then consider the os as "macOS" and return the message as "<Person name> is using macOS..."
 - a. Apple, apple, <space> Apple
5. If user provides any of the below values as os, then consider the os as "Windows" and return the message as "<Person name> is using Windows..."
 - a. <space>Windows, Windows, windows<space>
6. if user provides any other values as os, then return the message as ""<Person name> is using other OS..."

Exercise - 2

Test your function with the following input

1. "Rob"

Expected - Error: One or more vlaues are missing...

2. "Rob", 123

Expected - Error: Only string values are accepted, but received values are Rob and 123

3. "Rob", "Apple"

Expected - Hello Rob, you have provided apple as your operating system...

Rob is using macOS...

4. "Bob", "Windows"

Expected - Hello Bob, you have provided Windows as your operating system...

Bob is using Windows...

5. "John" "Linux"

Expected - Hello John, you have provided Linux as your operating system...

John is using other OS...

Exercise - 3

In your project root, under /data dir create a sub-folder called testFiles. Within test files, create different types of files (e.g. 3 text files, 2 json files, 3 csv files).

- a. Create a function that gets all files from testFiles directory and print the results in the console.
- b. Implement error handling so it can handle different scenario. Test your function with
 - i. Incorrect dir
 - ii. No files in the testFiles dir
 - iii. Correct path and files exists -> prints all the file names

Exercise - 4, 5, 6 (String Manipulation)

4. Print the following message in the console. Use single quote, double quote, and template literals

a. *I'm doing well*

5. In a given string, replace vowels with (-) and return the modified string

a. Given string: 'BYucze5zJlJOaJZTeXI8oWMeo87VeJqC'

b. Expected: BY-cz-5zJlJO-JZT-XI8-WM--87V-JqC

6. Convert SOUTH AFRICA to South Africa

Exercise - 7

Create an object from an given array, where the property name/key will be array's each element and value will be its equivalent number

- a. (e.g. {one: 1}) and print the output in the console
- b. `let arr = ["one", "two", "three", "four", "five"]`
- c. expected: { one: 1, two: 2, three: 3, four: 4, five: 5 }

Exercise - 8

Check if the value "JS" exists in the given array and print *Found* or *Not Found* accordingly

- a. Try this with with and without using Array's method
- b. `let arr = ["Java", "JS", "Python", "C#"]`

Exercise - 9, 10

9. Get only truthy value from given array and print the output

a. `let arr = ["0", 0, "", "one", "two", NaN, "three"]`

10. Convert all given string version of numbers into actual number array

a. `let arr = ["1", "2", "3", "4", "5", "six"];`

b. `expected: [1, 2, 3, 4, 5]`

Solutions

Refer to github page for sample solutions: <https://github.com/copeautomation/js-essential-for-automation>

Thank you