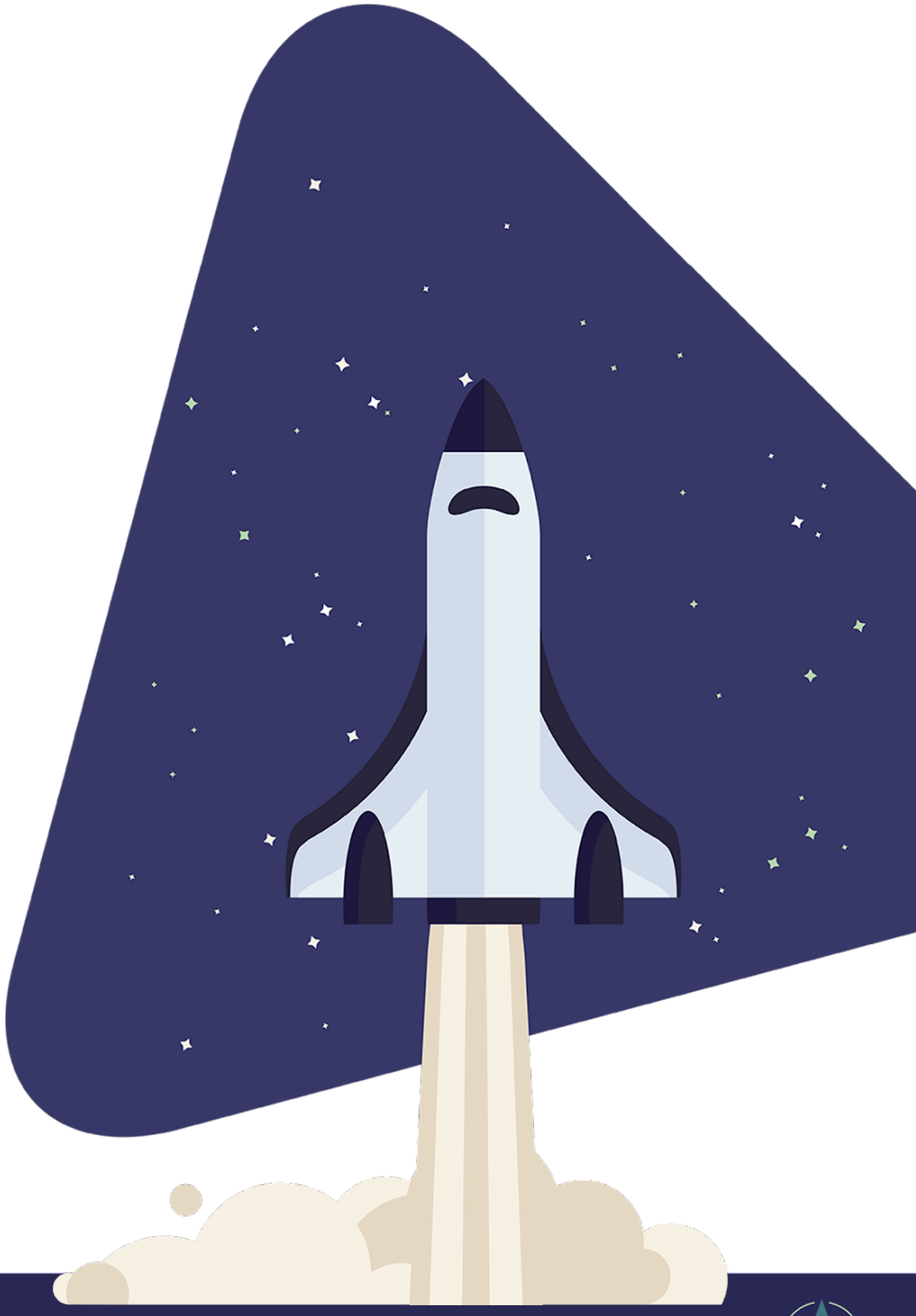
Become a Software Developer

Problem 04



Description of the problem to be solved

The problems of the fourth week will be divided into two activities:

1. On the one hand, considering the feedback provided by the tutors, they should improve what was delivered in the weekly problem 3 by implementing what was corrected.
2. On the other hand, they should perform the JavaScript exercises provided at the end of this document, applying the concepts learned from the asynchronous material.
   Each exercise block should be saved in a new JS file inside a folder called "exercises" and this folder should be saved in the new repository. All the Javascripts files should be linked to the "index.html" file through the <script> tag. The structure of the new repository should look like this:

```
index.html
exercises/
    01_variables_and_operators.js
    02_strings.js
    03_arrays.js
    04_if_else.js
    05_for.js
    06_functions.js
```

Steps to follow:

1) JavaScript:

   a) Create a new folder and inside it create a file in which the answer to the proposed JavaScript exercises will be included.

   b) Use Git Init, on the folder containing the file with the answers, to start Git.

   c) Create a new repository on GitHub and use Git Remote Add to link it.

   d) Solve the exercises that are part of the weekly problem.

   e) Make commits with the progress and upload the code to the new repository created in step c.

   f) Respond to Classroom Weekly Problem 04 with the Github link to the repository created before the submission deadline.

2) Responsive:

   a) Correction of the work handed in on the problem of week 03, based on the feedback received from the tutors.

   a) Make commits with the progress and upload the code in the week 03 repository.

b) Submit the Github Pages link of the week 03 repository as a response to the weekly issue.

It will be evaluated:

1. Solution of the problem; when opening the HTML in a web browser, the answers to the JavaScript exercises must be displayed correctly in the console.
2. Correct use of variables, operators, conditionals, iteration loops and functions to solve the different exercises.
3. Correction of all the points marked by the tutor for the problem of week 03.
4. Neatness and consistency of written code. All code and comments must be written in English.
5. Correct use of Git, Github, Readme document.
6. On-time delivery.
7. Active participation with the work group, asking questions to tutors and helping colleagues.

Weekly Problem Exercises

Description

Each of the following exercises must be solved and uploaded to a new Github repository.

Please solve them in the order given and make a commit for each exercise solved.

Add the statement of each exercise as a comment in the solution of the exercise.

List of exercises:

1) Variables and Operators

   a) Create two numeric variables and use the sum operator to store the sum value of both numbers in a 3rd variable.

   b) Create two String variables and concatenate them, saving the result in a 3rd variable.

   c) Create two String variables and add the length of each variable (number of letters of the string) saving the result of the sum in a 3rd variable (use length).

2) Strings

   a) Create a string variable with at least 10 characters and convert all text to uppercase (use toUpperCase).

   b) Create a string variable with at least 10 characters and generate a new string with the first 5 characters saving the result in a new variable (use substring).

   c) Create a string variable with at least 10 characters and generate a new string with the last 3 characters saving the result in a new variable (use substring).

   d) Create a string variable with at least 10 characters and generate a new string with the first letter in uppercase and the others in lowercase. Save the result in a new variable (use substring, toUpperCase, toLowerCase and the + operator).

   e) Create a string variable with at least 10 characters and some white space. Find the position of the first blank space and store it in a variable (use indexOf).

   f) Create a string variable with at least 2 long words (10 characters and some space in between). Use the methods from the previous exercises to generate a new string that has the first letter of both words in uppercase and the other letters in lowercase (use indexOf, substring, toUpperCase, toLowerCase and the + operator).

3) Arrays

   a) Given the following array: ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"] show by console the months 5 and 11 (use console.log).

   b) Sort the months array alphabetically and display it by console (use sort).

   c) Add an element to the beginning and end of the array (use unshift and push).

   d) Remove an element from the beginning and end of the array (use shift and pop).

   e) Invert the array order (use reverse).

   f) Join all elements of the array into a single string where each month is separated by a hyphen - (use join).

   g) Create a copy of the months array containing May through November (use slice).

4) If Else

   a) Create a random number between 0 and 1 using the Math.random() function, if the value is greater than or equal to 0.5 display an alert with the message "Greater than 0.5" and otherwise an alert with the message "Lower than 0.5".

b) Create an "Age" variable that contains an integer between 0 and 100 and displays the following alert messages:

   i) "Bebe" if the age is less than 2 years;

   ii) "Child" if age is between 2 and 12 years;

   iii) "Teenager" between 13 and 19 years old;

   iv) "Young" between 20 and 30 years old;

   v) "Adult" between 31 and 60 years of age;

   vi) "Older adult" between 61 and 75 years of age;

   vii) "Elderly" if over 75 years of age.

5) For

   a) Create an array containing 5 words and traverse that array using a JavaScript for loop to display an alert using each of the words.

   b) To the previous array convert the first letter of each word to uppercase and display an alert for each modified word.

   c) Create a variable called "sentence" that has an empty string, then run through the array in point a) with a for loop to store each word in the sentence variable. At the end display a single alert with the complete string.

   d) Create an empty array with a for loop of 10 repetitions. Fill the array with the number of the repetition, that is to say that at the end of the execution of the for loop there should be 10 elements inside the array, from number 0 to number 9. Display the final array in the browser console (use console.log).

6) Functions

   a) Create a sum function that receives two numeric values and returns the result. Execute the function and save the result in a variable, displaying the value of that variable in the browser console.

   b) To the sum function above, add a validation to check if any of the parameters is not a number, display a warning that one of the parameters has an error and return the NaN value as the result.

   c) Create a validate integer function that receives a number as a parameter and returns true if it is an integer.

d) To the function sum of the exercise 6b) add a call that validates that the numbers are integers. In case there are decimals, display an alert with the error and return the number converted to integer (rounded).

e) Convert the validation of exercise 6d) into a separate function and call it inside the sum function testing that everything still works the same.