Exercise 1

The login and registration form is a very popular and integral part of many websites. Now you are working with it in JavaScript.

Let's imagine that on our site there is a user whose login is "the\_best\_user" and password is "12345678". Store these values in variables and name them existsUserLogin and existsUserPassword respectively.

You need to use the prompt function to display 2 modal windows on the screen. The 1st will be with the inscription "Enter login", the 2nd - "Enter password". Store the user-entered values in the userLogin and userPassword variables. Also note that the user summary must not contain extra spaces at the beginning and end of the line. use trim method to remove extra spaces.

If the entered username and password match the existing data in existsUserLogin and existsUserPassword, then display in the modal window with the message alert "Welcome, userLogin!" (replace userLogin with the user's login). Otherwise, display the message "Login and (or) Password entered incorrectly!".

Note: Never use the non-strict "==" comparison in web application development. It is the initiator of a large number of bugs, as it performs type conversion. combine exclusively strict comparison “===”. It does not convert types and assumes the possibility of errors.

In the task, also tag the strict comparison “===”.

Task 2

This code asks for the names of new students 3 times and, if a name has been entered, the student is greeted with the message "Welcome, new Student!" (newStudent - student's name).

You need to rewrite this loop for a while and do while loop so that the code execution logic and the result remain unchanged.

for (let i = 0; i < 3; i += 1) {

let newStudent = prompt('Enter the name of the new student!');

newStudent = newStudent.trim();

if(newStudent) {

alert(`Welcome ${newStudent}!`)

}

}

Task 3

Let's continue with cycles.

In the future, you are given the number 100. You need to calculate the amount from 0 to 100 (or vice versa, from 100 to 0). You end up with a value of 5050. Store the total as a variable amount and display it on the screen with an alert.

Task 4

Have you ever taken tests? Your answer is yes, as they were repeated. Now almost all tests have been transferred to online formats and posted on websites developed by web developers. Release a short math test for elementary school students.

There are only 5 questions in the test:

What's 2 2?

How much will 2 \* 2 be?

Petya had 5 apples. He ate 3 of them and gave 1 to a friend. How many apples does Peter have left?

Masha had 10 sweets. She ate 2 and gave 1 to a friend. After that, mom gave Masha 5 more sweets. How many sweets did Masha have left in the end?

What is 2 + 2 \* 2?

You need to help primary school students, so they need a test that confirms their knowledge of mathematics.

Each question and pre-prepared correct answer (answers must be a data type) Questions are output using a function hint, and also store the user's input in certain variables.

If the answer was given correctly, then display the message "Correct answer" with an alert, otherwise - "Incorrect answer". Even if the answer was given incorrectly, the user should still be redirected to the next question.

You also need to keep a record of correct and incorrect answers. Store this data in the variables correct answers and incorrect answers, respectively. After running the test with an alert, display the message “End of test! Correct answers - correct answers; Wrong answers are wrong answers. (replace correct answers and incorrect answers with totals).

Note: Note that the data type hint returns a string, and the values to be found in the answer options is the data type number.

Task 5

Imagine that we are developing a website for a restaurant that has a discount program. The restaurant calculates the amount of payment for each client and encourages the most frequent visitors.

At the moment, the restaurant has a client named Igor, who spent $ 110 in the entire time in this institution. Store this data in the clientName and clientSpentForAllTime variables.

The discount program works as follows:

If the client spent from $100 to $300, then a 10% discount;

If the client spent from $300 to $500, then a 20% discount;

If the client spent from $500, then a 30% discount;

A client named Igor decided to visit the restaurant again. As we can see, he is entitled to a 10% discount since he spent $110. Now Igor is purchased at the establishment for $25 and he needs to be given a discount. Store the value 25 in the clientSpentToday variable.

First, implement the logic for discounts. Use if else and logical operators. Store the final discount in the discount variable. Display in the modal window via alert the text “You are given a discount in discount%!” (replace discount with discount percentage).

After that, it is necessary to issue an invoice for payment to a client named Igor (taking into account the discount). First, update the clientSpentForAllTime variable. Then display in the modal window via alert the message “Thank you, clientName! Payable clientSpentToday$. For all the time in our restaurant you spent clientSpentForAllTime$.” (replace clientName, clientSpentToday and clientSpentForAllTime with appropriate variable values)

Task 6

Congratulations! You have developed the logic for giving a discount to restaurant customers. You've optimized a big task, because earlier discounts were calculated manually.

You have already worked with the predefined data clientName, clientSpentToday, and clientSpentForAllTime. But the restaurant has a lot of customers and everyone buys dishes at different prices and everyone needs to provide different discounts. Therefore, now we will ask the user to enter the desired data in the text field.

Let's use the prompt function already known to us. Set each of the following variables clientName, clientSpentToday, and clientSpentForAllTime to the value that the user enters in the text box. The resulting clientSpentToday and clientSpentForAllTime data must be of the number data type.

For clientName, the message in the prompt should be “Enter the client's name”, clientSpentToday - “How much did the client spend today?”, clientSpentForAllTime - “How much did the client spend all time?”.

If the user entered incorrect data for clientSpentToday and clientSpentForAllTime (for example, “hello”), then display the text in the modal window “The amount that the client spent all time and spent today must be a number! Please reload the page to try again." If the data is incorrect, do not allow the user to go further to the discount calculation. Use if else to implement the given logic