Example-1

Create a function registerUser() that uses pop-up boxes to gather and validate user input for a new user registration.

The function should:

1. Use a while loop to repeatedly prompt the user for a username using the prompt() pop-up box. The loop should continue to run as long as the user's input is an empty string, null, or undefined.
2. Once a valid username is provided, use another while loop to prompt the user for a password. This loop should continue until the user provides a password that is at least 8 characters long.
3. After a valid password is entered, use the confirm() pop-up box to ask the user,
   Are you sure you want to register with this username and password?
4. If the user clicks Cancel on the confirmation box, use alert() to display a message
   Registration canceled.
   and end the function.
5. If the user clicks OK, use a template literal to display a final alert() message confirming the successful registration, showing both the username and the password.

Example-2

Write a JavaScript program that first uses the alert box to display a welcome message to the user (Example: Welcome User). Then, use the confirm box to ask the user if they want to continue.

1. If the user clicks OK, proceed to ask for their name using the prompt box and display a personalized greeting
   (Example: If user enters Jack, then display Hello Jack).
   After displaying the greeting using alert box, prompt the user to enter their age.
   Based on the user's age, use a conditional statement to check if they are 18 or older.
   If the user is 18 or older, display an alert saying
   You are an adult!
   If the user is under 18, display an alert saying
   You are a minor!.
2. If the user clicks Cancel in the confirm box at the beginning, display an alert saying Goodbye.

Example-3

Write a function gradeStudents that takes an array of student objects (each with a name and FSD marks property).

The function should:

1. Iterate through the student array.
2. For each student, it should use the prompt() pop-up box to ask the user to enter FSD marks.

The prompt message should clearly state the student's name (e.g., Enter FSD marks for Alice:).

3. It should validate the user input. If the input is not a valid number (use isNaN()), display an alert() message
   Invalid grade. Please enter a number.
   and continue to the next student without updating the current one.
4. If the input is valid, update the student's FSD marks property with the new value.
5. Finally, after the loop finishes, it should use alert() to display a summary of the FSD marks for all students in a clear, multi-line format using template literals.

Example-4

Given the following HTML snippet:

```
<h1 id="main-title">Hello DOM!</h1>
```

Write a JavaScript function that performs the following tasks:

1. Selects the h1 element
2. Changes its text content to **DOM Manipulation is Fun!** using innerText.
3. Changes its text color to green using the style property.

Example-5

Given the following HTML:

```
<p class="description">Item 1</p>
<p class="description">Item 2</p>
<p class="description">Item 3</p>
```

Write a JavaScript function that:

1. Selects all elements with the class **description**
2. For each element, changes to **Updated!** inside strong tag.
3. Also, adds a title attribute to each element with the value **This item has been updated.**

Example-6

Create a form with two fields: username and email, along with a submit button.

1. When the form is submitted, prevent the default form submission and display an alert box with the message
   Form submitted.
2. When the user makes changes to any field and then leaves it (loses focus), the background color of that field should change to yellow.
3. While user clicks on input field, change background color to pink (only for username field).
4. While the user is typing in a field, the input text should appear in red color.
5. Implement this using only event attributes (oninput, onchange, onsubmit) in HTML, without using addEventListener.

Example-7

Create a form with two fields (username and email) and a submit button. Implement the following:

1. While typing in a field, the text should appear in red (oninput).
2. When the user leaves a field after making changes, its background should change to yellow (onchange).
3. When the user hovers (onmouseover) over the form, the border should turn green.
4. When the mouse leaves the form (onmouseout), the border should return to black.
5. Add a separate button that toggles the form's background color between white and lightblue (onclick).
6. When the form is submitted, prevent default submission and display an alert saying **Form submitted successfully!** (onsubmit).

Example-8

Write a JavaScript program that displays an alert message **"welcome user"** when the page loads. Then, display a confirmation box asking **"Do you want to continue?"**

- If the user clicks **OK**, prompt the user to enter two numbers.
- Calculate the sum of the two numbers.
- Display the result in an alert box.
- If the user clicks **Cancel**, display an alert message **"bye bye"**.

Example-9

Write a single function calculateDiscount that meets the following requirements: It takes a basePrice as its first parameter. It accepts an optional discount percentage as a second parameter, which should default to 10. It uses a Rest Parameter to accept any number of additional prices to be included in the calculation. The function should return the sum of all prices passed to it (including the basePrice and additional prices), after applying the discount to the total.

Example-10

Design a login form using HTML & JavaScript with following validations on username and password fields.

1. Password length must be 6 to 12 characters
2. Username should not start with _, @ or number.
3. Both should not be blank.

Example-11

Design a login form with username and password option using HTML and JavaScript. Perform following validations.

(i) Username field: minimum length 6 characters, it should not have any special character or digits other than _

(ii) Password field: minimum length 8 characters and maximum length 12 characters, it must contain at least one digit and at least one special character from the set {*, #, _}

Example-12

Design a registration form with username and password and email fields using HTML and JavaScript. Perform following validations.

(i) Username field: minimum length 5 and maximum length 10 characters, it should not start with special characters

(ii) Password field: it must contain at least one digit and at least one special character

(iii) Email field: it must follow proper email syntax (for ex., xyz.pqr@mnp.co.in)

Example-13

Create a small HTML page where:

You store two fields in sessionStorage:

username = lj

email = lj@example.com

Retrieve and display the stored username in the console.

Remove only the email field from sessionStorage, then print the remaining data in the console.

Example-14

Create a small HTML page where:

You store three fields in localStorage:

product = Laptop

price = 50000

discount = 10

Retrieve and display the stored product in the console.

Remove the discount field from localStorage, then print the updated data in the console.

Example-15

Create a form with three fields:

product, price, discount and a Save button.

Requirements:

1. When the user clicks Save, store the entered values into localStorage.
2. After saving, display a message: Data saved in local storage!
3. Add another button Remove Discount that removes only the discount field from localStorage.
4. After removal, display the remaining data in the console.