how to form Basic form validation (email, password, phone number validation) using java script for learning

To create basic form validation for an email, password, and phone number using JavaScript, you can use regular expressions (regex) for validation and simple JavaScript functions to check the input fields. Here's a step-by-step guide to form validation in a basic HTML form with JavaScript.

**1. HTML Form Setup**

Start by creating a basic HTML form with fields for email, password, and phone number:

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Form Validation Example</title>

<style>

.error {

color: red;

font-size: 12px;

}

</style>

</head>

<body>

<h2>Signup Form</h2>

<form id="signupForm" onsubmit="return validateForm()">

<label for="email">Email:</label>

<input type="email" id="email" name="email">

<span id="emailError" class="error"></span>

<br><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password">

<span id="passwordError" class="error"></span>

<br><br>

<label for="phone">Phone Number:</label>

<input type="text" id="phone" name="phone">

<span id="phoneError" class="error"></span>

<br><br>

<input type="submit" value="Submit">

</form>

<script src="validation.js"></script>

</body>

</html>

**2. JavaScript Form Validation (validation.js)**

Now let's write the JavaScript for validating the email, password, and phone number.

javascript

Copy code

function validateForm() {

// Clear previous error messages

clearErrors();

let isValid = true;

// Get form elements

const email = document.getElementById("email").value;

const password = document.getElementById("password").value;

const phone = document.getElementById("phone").value;

// Validate Email

if (!validateEmail(email)) {

document.getElementById("emailError").textContent = "Please enter a valid email address.";

isValid = false;

}

// Validate Password

if (!validatePassword(password)) {

document.getElementById("passwordError").textContent = "Password must be at least 6 characters long.";

isValid = false;

}

// Validate Phone Number

if (!validatePhone(phone)) {

document.getElementById("phoneError").textContent = "Please enter a valid phone number (e.g., 123-456-7890).";

isValid = false;

}

return isValid; // If any validation fails, the form will not submit

}

// Clear previous error messages

function clearErrors() {

document.getElementById("emailError").textContent = "";

document.getElementById("passwordError").textContent = "";

document.getElementById("phoneError").textContent = "";

}

// Email validation function

function validateEmail(email) {

// Regular expression for a basic email validation

const emailPattern = /^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/;

return emailPattern.test(email);

}

// Password validation function

function validatePassword(password) {

// Check if password is at least 6 characters long

return password.length >= 6;

}

// Phone number validation function

function validatePhone(phone) {

// Regular expression for phone number validation (e.g., 123-456-7890 or (123) 456-7890)

const phonePattern = /^[\d\-\(\) ]{10,15}$/;

return phonePattern.test(phone);

}

**3. Explanation of Code**

* **HTML Form:**
  + The form contains input fields for email, password, and phone number.
  + Each field has a corresponding span element to display error messages.
  + The form uses onsubmit="return validateForm()", which calls the validateForm() JavaScript function before submitting the form.
* **JavaScript Functions:**
  + validateForm(): This function is called when the form is submitted. It checks all the input fields and displays error messages if the validation fails. It returns false if validation fails, preventing the form from being submitted.
  + **Email Validation (validateEmail)**:
    - Uses a regular expression (/^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/) to ensure the email is in the correct format.
  + **Password Validation (validatePassword)**:
    - Checks if the password length is at least 6 characters.
  + **Phone Number Validation (validatePhone)**:
    - Uses a regular expression (/^[\d\-\(\) ]{10,15}$/) to validate a phone number format. This allows dashes, spaces, and parentheses, and ensures the number is between 10 and 15 characters long.
* **Error Messages:**
  + Each input field has an associated span tag (id="emailError", etc.) where error messages are displayed if validation fails.

**4. How It Works**

1. When the user submits the form, the validateForm() function is triggered.
2. Each input field is validated using the corresponding validation function (validateEmail(), validatePassword(), validatePhone()).
3. If any of the validation checks fail, an error message is displayed beside the input field, and the form submission is prevented (return false).
4. If all the validations pass, the form is submitted as usual.

**5. Example Output (With Invalid Inputs)**

1. **Email:** If the user enters invalid-email, the error message will say "Please enter a valid email address."
2. **Password:** If the user enters a password shorter than 6 characters, the message will say "Password must be at least 6 characters long."
3. **Phone:** If the user enters 1234567890 without any dashes, spaces, or parentheses, the message will say "Please enter a valid phone number."

**Conclusion**

This basic form validation approach using JavaScript ensures that the form data is correct before submission. The regular expressions used here are quite simple, but you can make them more complex depending on the requirements. This example is great for beginners to understand how to validate common input fields in a form.

Top of Form



Bottom of Form

ChatGPT can make mistakes. Check important info.

?