

JavaScript Practice Lab: Form Validation and DOM Manipulation

Lab Objective

By the end of this lab, students will be able to:

1. Validate form inputs using JavaScript.
2. Manipulate the DOM to display validation messages.
3. Create a dynamic user interface based on user input.

Prerequisites

Students should have a basic understanding of HTML, CSS, and JavaScript fundamentals.

Lab Setup

1. Create a new folder named `form-validation-lab`.
2. Inside this folder, create three files: `index.html`, `styles.css`, and `script.js`.

Lab Instructions

Part 1: HTML and CSS Setup

1. **index.html:** Set up the basic HTML structure with a form.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Form Validation Lab</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Form Validation Lab</h1>
  <div id="form-container">
    <form id="registrationForm">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name">
      <span class="error" id="nameError"></span>
      <br>
      <label for="email">Email:</label>
      <input type="email" id="email" name="email">
      <span class="error" id="emailError"></span>
      <br>
      <label for="password">Password:</label>
      <input type="password" id="password" name="password">
```

```

        <span class="error" id="passwordError"></span>
        <br>
        <button type="submit">Register</button>
    </form>
    <div id="message"></div>
</div>
<script src="script.js"></script>
</body>
</html>

```

2. **styles.css**: Add some basic styles.

```

body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

#form-container {
    background-color: #fff;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

label {
    display: block;
    margin-top: 10px;
}

input {
    width: 100%;
    padding: 8px;
    margin-top: 5px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

.error {
    color: red;
    font-size: 0.9em;
}

button {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #28a745;
    color: white;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

```

```

button:hover {
    background-color: #218838;
}

#message {
    margin-top: 20px;
    font-size: 1.1em;
    color: #28a745;
}

```

Part 2: JavaScript for Form Validation and DOM Manipulation

3. **script.js:** Implement form validation and DOM manipulation.

```

document.getElementById('registrationForm').addEventListener('submit'
, function(event) {
    event.preventDefault();
    clearErrors();
    let isValid = validateForm();

    if (isValid) {
        displayMessage('Registration successful!', 'success');
    }
});

function validateForm() {
    let isValid = true;

    // Validate name
    let name = document.getElementById('name').value;
    if (name.trim() === '') {
        showError('nameError', 'Name is required');
        isValid = false;
    }

    // Validate email
    let email = document.getElementById('email').value;
    if (email.trim() === '') {
        showError('emailError', 'Email is required');
        isValid = false;
    } else if (!isValidEmail(email)) {
        showError('emailError', 'Invalid email format');
        isValid = false;
    }

    // Validate password
    let password = document.getElementById('password').value;
    if (password.trim() === '') {
        showError('passwordError', 'Password is required');
        isValid = false;
    } else if (password.length < 6) {
        showError('passwordError', 'Password must be at least 6
characters');
        isValid = false;
    }

    return isValid;
}

```

```

}

function showError(elementId, message) {
    document.getElementById(elementId).textContent = message;
}

function clearErrors() {
    document.getElementById('nameError').textContent = '';
    document.getElementById('emailError').textContent = '';
    document.getElementById('passwordError').textContent = '';
}

function displayMessage(message, type) {
    let messageElement = document.getElementById('message');
    messageElement.textContent = message;
    messageElement.className = type;
}

function isValidEmail(email) {
    // Simple email validation regex
    let re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return re.test(email);
}

```

4. Test the setup by opening `index.html` in a web browser, filling out the form, and submitting it to see the validation messages.

Additional Challenges

1. Add an input field for phone number and validate it to ensure it contains only digits and is exactly 10 characters long.
2. Add a "Confirm Password" field and validate that it matches the "Password" field.
3. Enhance the validation feedback to include icons or colors to indicate success and error states.
4. Implement real-time validation: display validation messages as the user types.