

HTML Exercises Guide - Forms

Lesson Objectives

1. Remembering

- Define what an HTML form is and its role in web development.
- Recall different HTML form elements such as `<form>`, `<input>`, `<label>`, `<select>`, `<textarea>`, `<button>`.

2. Understanding

- Explain the difference between **GET** and **POST** methods.
- Describe the purpose of attributes like `name`, `id`, `value`, `placeholder`, and `required`.
- Discuss why labels improve accessibility and usability.

3. Applying

- Use different `<input>` types (text, password, email, number, checkbox, radio, date, etc.) in practice exercises.
- Apply `label` elements correctly by linking them with inputs.
- Build a simple contact or login form using form elements.

4. Analyzing

- Compare and contrast when to use text fields vs. radio buttons vs. dropdowns.
- Distinguish between required vs. optional fields and their effect on user experience.
- Examine how form data is structured before submission.

5. Evaluating

- Assess whether a form is user-friendly and accessible.

- Identify potential issues in a form design (e.g., missing labels, unclear placeholders).

6. Creating

- Design and implement a fully functional HTML form (e.g., registration form, feedback form).
- Prepare forms for integration with JavaScript and jQuery by ensuring proper structure and attribute usage.

What is an HTML Form?

An **HTML form** is a way for users to enter and send information to a server (or for now, just inside the page).

Think of it as a **digital paper form**.

Forms usually include:

- **Inputs** (text, password, email, number, etc.)
- **Labels** (descriptions for inputs)
- **Buttons** (submit, reset)
- **Selects, checkboxes, radios** (choices)
- **Textarea** (long text entry)

Basic Form Structure

```
<form action="submit.php" method="POST">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>

  <label for="password">Password:</label>
  <input type="password" id="password" name="password" required>

  <button type="submit">Login</button>
</form>
```

Key Parts:

- `<form>` → wraps everything.
- `action="submit.php"` → where data goes (server page, file, API).
- `method="POST"` → how data is sent (GET or POST).
- `name="username"` → identifies the input's data when submitted.
- `required` → ensures the user fills it before submitting.

Form Methods: GET vs POST

- **GET** → Data appears in URL (e.g., `?username=John`). Good for search forms.
- **POST** → Data hidden from URL. Used for logins, sensitive info.

Input Types

Forms can collect different kinds of data using `type`.

```
<input type="text" name="fullname" placeholder="Enter your name">
<input type="email" name="email" placeholder="Enter your email">
<input type="password" name="password" placeholder="Enter password">
<input type="number" name="age" min="1" max="120">
<input type="date" name="birthday">
<input type="checkbox" name="subscribe"> Subscribe to newsletter
<input type="radio" name="gender" value="male"> Male
<input type="radio" name="gender" value="female"> Female
<select name="country">
  <option value="ph">Philippines</option>
  <option value="us">United States</option>
</select>
<textarea name="message" rows="4"></textarea>
```

Labels & Accessibility

Always pair inputs with `<label>` for better usability:

```
<label for="email">Email Address</label>
<input type="email" id="email" name="email">
```

Buttons

```
<button type="submit">Submit</button>
<button type="reset">Reset</button>
<button type="button">Just a Button</button>
```

- **submit** → sends the form
- **reset** → clears all inputs
- **button** → does nothing by default (use JS later)

Putting It Together (Mini Exercise – Contact Form)

```
<form action="#" method="POST">
  <h2>Contact Us</h2>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" placeholder="Your name"
required>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" placeholder="Your
email" required>
  <label for="message">Message:</label>
  <textarea id="message" name="message" rows="4" required></textarea>

  <button type="submit">Send</button>
</form>
```

Step-by-Step HTML Forms → JS/jQuery Exercises

Exercise 1: Create a Basic Login Form (HTML Only)

```
<form>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required><br><br>

  <label for="password">Password:</label>
  <input type="password" id="password" name="password"
required><br><br>

  <button type="submit">Login</button>
</form>
```

Exercise 2: Add Email Input & Placeholder

👉 **Goal:** Introduce new input types and usability.

```
<form>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" placeholder="Enter your
email">
</form>
```

Exercise 3: Add Radio Buttons & Checkboxes

👉 **Goal:** choice-based inputs.

```
<p>Choose your role:</p>
<input type="radio" id="student" name="role" value="student">
<label for="student">Student</label>
<input type="radio" id="teacher" name="role" value="teacher">
<label for="teacher">Teacher</label>

<p>Select your interests:</p>
<input type="checkbox" id="html" name="interest" value="HTML">
<label for="html">HTML</label>
<input type="checkbox" id="css" name="interest" value="CSS">
<label for="css">CSS</label>
```

Exercise 4: Add a Dropdown (Select Menu)

👉 **Goal:** `<select>` element.

```
<label for="country">Country:</label>
<select id="country" name="country">
  <option value="ph">Philippines</option>
  <option value="us">United States</option>
  <option value="jp">Japan</option>
</select>
```

Exercise 5: Add a Textarea

👉 **Goal:** Collect longer user input.

```
<label for="message">Your Message:</label><br>

<textarea id="message" name="message" rows="4" cols="40"></textarea>
```

Exercise 6: Form Validation with HTML Attributes

👉 **Goal:** Show built-in validation.

```
<input type="text" name="username" required minlength="3"
maxlength="10">

<input type="email" name="email" required>
```

Exercise 7: Add JS Alert on Submit (Native JavaScript)

👉 **Goal:** Introduce `onsubmit` event.

```
<form id="loginForm">

  <input type="text" id="username" required>

  <button type="submit">Submit</button>

</form>

<script>

document.getElementById("loginForm").addEventListener("submit",
function(event) {

  event.preventDefault();

  alert("Form submitted!");

});

</script>
```

Exercise 8: Show Live Character Count (Native JavaScript)

👉 **Goal:** Teach input events.

```
<textarea id="bio" maxlength="100"></textarea>

<p id="count">0/100</p>

<script>

const bio = document.getElementById("bio");
const count = document.getElementById("count");

bio.addEventListener("input", function() {
    count.textContent = bio.value.length + "/100";
});

</script>
```

Exercise 9: Toggle Password Visibility (Native JavaScript)

👉 **Goal:** Manipulate input attributes.


```
<input type="password" id="pwd">

<button type="button" onclick="togglePwd()">Show/Hide</button>

<script>

function togglePwd() {

    const pwd = document.getElementById("pwd");

    pwd.type = pwd.type === "password" ? "text" : "password";

}

</script>
```

Exercise 10: jQuery Version of Submit Alert

```
<form id="myForm">

    <input type="text" required>

    <button type="submit">Submit</button>

</form>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script>

$("#myForm").submit(function(event){

    event.preventDefault();

    alert("Form submitted with jQuery!");

});

</script>
```

Exercise 11: jQuery Show/Hide Password

👉 **Goal:** Compare with Native JS.

```
<input type="password" id="pwd">

<button id="toggle">Show/Hide</button>


<script>

$("#toggle").click(function(){

    let pwd = $("#pwd");

    pwd.attr("type", pwd.attr("type") === "password" ? "text" :
"password");

});

</script>
```

Exercise 12: jQuery Live Character Count

👉 **Goal:** Practice jQuery events.

```
<textarea id="bio" maxlength="100"></textarea>

<p id="count">0/100</p>


<script>

$("#bio").on("input", function(){

    $("#count").text($(this).val().length + "/100");

});

</script>
```

Exercise 13: Validate Empty Fields with jQuery

👉 **Goal:** Introduce custom validation.

```
<form id="login">

  <input type="text" id="user" placeholder="Username"><br>

  <input type="password" id="pass" placeholder="Password"><br>

  <button type="submit">Login</button>

</form>


<script>

$("#login").submit(function(e){

  e.preventDefault();

  if($("#user").val() === "" || $("#pass").val() === ""){

    alert("Please fill in all fields");

  } else {

    alert("Login successful!");

  }

});

</script>
```

Exercise 14: Add CSS Highlight for Errors (with jQuery)

👉 **Goal:** Introduce UI feedback.

```
<style>

.error { border: 2px solid red; }

</style>


<form id="formCheck">

    <input type="text" id="name" placeholder="Name">

    <button type="submit">Submit</button>

</form>


<script>

$("#formCheck").submit(function(e){

    e.preventDefault();

    if($("#name").val() === ""){

        $("#name").addClass("error");

    } else {

        $("#name").removeClass("error");

        alert("Form submitted!");

    }

});

</script>
```

Exercise 15: Build a Full Registration Form (HTML + jQuery)

👉 **Goal:** Combine everything learned.

Features: Username, Email, Password, Confirm Password, Gender, Interests, Submit.

✅ **You must:**

- Use labels, inputs, radio, checkbox, select.
- Add validation for empty fields.
- Add live character count on bio.
- Add password toggle.

Save it as *registration.html* and put it in your github repository, integrate it with your group system project later on.