

Introduction to HTML Forms

Emmanuel BRUNO

Université de Toulon, LIS UMR CNRS 7020

emmanuel.bruno@univ-tln.fr

2026-10-05

Table des matières

1	Creating Interactive Forms with HTML . .	1
2	Basic Form Elements - Text and Password Fields	1
3	Basic Form Elements - Checkboxes and Radio Buttons	2
4	Advanced Form Elements	2
5	Advanced Form Elements	2
6	Advanced Form Elements	2
7	Form Validation	3
8	Form Styling	3
9	Accessibility in Forms	3
10	Example Form with Labels and Placeholders	4

1 Creating Interactive Forms with HTML

- **Purpose of Forms:**
 - Collect user input and data for various applications.
 - Facilitate user interactions on websites.
- **HTML Form Elements:**
 - Provide a variety of elements to create different types of forms.
 - Include text fields, checkboxes, radio buttons, dropdowns, and more.
- **Best Practices:**
 - Design forms for usability and accessibility.
 - Ensure clear labels and instructions.

- Validate input on both client-side and server-side.

- Style forms to enhance visual appeal and user experience.

- **Overview:**

- Explore common form elements.
- Learn best practices for form design and usability.

2 Basic Form Elements - Text and Password Fields

- **Text fields:** `<input type="text">`

- For single-line text input.

- Example:

```
<label for="name">Name:</label>
<input type="text" id="name"
name="name" placeholder="Enter
your name">
```

- **Password fields:** `<input type="password">`

- For secure password input.

- Example:

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

```
placeholder="Enter your  
password">
```

```
cols="50" placeholder="Enter your  
comments"></textarea>
```

3 Basic Form Elements - Checkboxes and Radio Buttons

- **Checkboxes:** `<input type="checkbox">`

- For multiple-choice options.
- Example:

```
<label for="subscribe">Subscribe  
to newsletter:</label>  
<input type="checkbox"  
id="subscribe" name="subscribe">
```

- **Radio buttons:** `<input type="radio">`

- For single-choice options.
- Example:

```
<label>Gender:</label>  
<input type="radio" id="male"  
name="gender" value="male">  
<label for="male">Male</label>  
<input type="radio" id="female"  
name="gender" value="female">  
<label for="female">Female</label>
```

4 Advanced Form Elements

- **Textarea:** `<textarea>`

- For multi-line text input.
- Example:

```
<label  
for="comments">Comments:</label>  
<textarea id="comments"  
name="comments" rows="4">
```

- **Select element:** `<select>`

- For dropdown lists.
- Example:

```
<label for="country">Country:</label>  
<select  
id="country" name="country">  
  <option value="usa">USA</option>  
  <option value="canada">Canada</option>  
  <option value="uk">UK</option>  
</select>
```

5 Advanced Form Elements

- **File input:** `<input type="file">`

- For file uploads.
- Example:

```
<label for="resume">Upload your  
resume:</label>  
<input type="file" id="resume"  
name="resume">
```

- **Date input:** `<input type="date">`

- For date selection.
- Example:

```
<label  
for="birthday">Birthday:</label>  
<input type="date" id="birthday"  
name="birthday">
```

6 Advanced Form Elements

- **Time input:** `<input type="time">`

- ▶ For time selection.
- ▶ Example:

```
<label
  for="appointment">Appointment
Time:</label>
<input
  type="time"      id="appointment"
  name="appointment">
```

7 Form Validation

- **Client-side validation:**
 - ▶ Validate data using HTML5 attributes and JavaScript before submitting.
 - ▶ Provides immediate feedback to users, improving user experience.
 - ▶ Reduces server load by catching errors early.
- **Server-side validation:**
 - ▶ Validate data on the server for security and reliability.
 - ▶ Ensures data integrity even if client-side validation is bypassed.
 - ▶ Essential for protecting against malicious input and ensuring accurate data processing.
- **Common validations:**
 - ▶ **Required fields:** Ensure that essential fields are not left empty.
 - ▶ **Email format:** Validate that the input matches a standard email format.
 - ▶ **Number range:** Ensure that numerical inputs fall within a specified range.
 - ▶ **Custom rules:** Implement specific validation rules tailored to the application's needs.

8 Form Styling

- **Use CSS to customize the appearance of forms:**
 - ▶ Enhance the visual appeal of forms by applying custom styles.

- ▶ Use CSS to control the layout, typography, colors, and spacing of form elements.
- **Consider typography, colors, spacing, and layout:**
 - ▶ **Typography:** Choose readable fonts and appropriate font sizes for form labels and inputs.
 - ▶ **Colors:** Use a color scheme that aligns with your brand and ensures good contrast for readability.
 - ▶ **Spacing:** Provide adequate spacing between form elements to prevent a cluttered appearance.
 - ▶ **Layout:** Organize form elements in a logical and user-friendly manner, using grids or flexbox for alignment.
- **Ensure the form is visually accessible and easy to use:**
 - ▶ **Accessibility:** Use high-contrast colors and ensure that form elements are easily distinguishable.
 - ▶ **Focus States:** Style focus states to make it clear which form element is currently active.
 - ▶ **Error Messages:** Style error messages to be noticeable and informative, guiding users to correct their input.
 - ▶ **Responsive Design:** Ensure the form is responsive and works well on various screen sizes and devices.

9 Accessibility in Forms

- **Use appropriate HTML elements and attributes:**
 - ▶ Use semantic elements like `<label>`, `<fieldset>`, and `<legend>`.
- **Provide alternative text for images:**
 - ▶ Ensure all images have descriptive `alt` attributes.
- **Use clear and concise labels:**
 - ▶ Ensure labels are descriptive and clearly associated with form elements.

- **Consider keyboard navigation:**
 - Ensure all form elements are accessible via keyboard.
 - Provide logical tab order and focus states.

10 Example Form with Labels and Placeholders

- **Complete Example:**

```
<form      action="/submit"
method="post">
  <label for="name">Name:</label>
    <input type="text" id="name"
name="name" placeholder="John Doe">

    <label for="email">Email:</label>
      <input
type="email" id="email" name="email"
placeholder="example@example.com">

    <label for="message">Message:</
label>
      <textarea id="message"
name="message" placeholder="Enter
your message here" rows="4"
cols="50"></textarea>

      <input type="submit"
value="Submit">
</form>
```