Forms in More Detail and Embedded Media

Collecting user input and enriching web pages with media

- A deeper look at HTML forms and user interaction
- A practical introduction to audio, video, images, and embedded content
- Focus: usability, structure, validation, and meaningful media use

Teaching Focus

A deeper look at HTML forms and user interaction

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Review of the Previous Lesson

Tables and Introduction to Forms

- We learned how tables organize information into rows and columns.
- We introduced the <form> element and basic input fields.
- We discussed action, method, labels, and simple validation.
- Today, we build on that foundation and make forms more realistic.

Teaching Focus

We learned how tables organize information into rows and columns.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Why Do We Use Forms?

- Forms collect information from users.
- They are used in login pages, registration pages, contact pages, search boxes, booking pages, and feedback pages.
- A form creates interaction between the user and the website.
- Without forms, a website can display information, but it cannot easily gather information.

- Login forms
- Contact forms
- Registration forms
- Feedback forms
- Search forms

Teaching Focus

Forms collect information from users.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Basic Form Structure Review

- A form starts with the <form> element.
- Inside the form, we place labels, input fields, menus, text areas, and buttons.
- The action attribute tells the browser where the data should go.
- The method attribute tells the browser how the data should be sent.
- A submit control sends the form.

HTML Example

```
<form action="/submit" method="post">
  <label>Name</label>
  <input type="text">
  <button type="submit">Send</button>
</form>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

How Form Submission Works

- The user enters data into inputs.
- The browser collects values by using the name attribute
- Each value is paired with its name attribute.
- The browser sends the data to the action URL.
- When the user clicks Submit, the browser sends the data to the address in action.
- The server receives and processes the request.

HTML Example

```
username=ali
email=ali@example.com
message=Hello
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Understanding action and method

- action = the destination URL for submitted data
- method = GET or POST
- method = the transmission style of the form data
- Together they define where data goes and how it travels.
- If action is missing, the form may submit to the current page.
- If method is missing, the browser usually uses GET by default.

HTML Example

```
<form action="contact.php" method="post">
  ...
</form>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

GET vs POST

- GET usually places the submitted data in the URL.
- GET is suitable for search, filtering, and non-sensitive requests.
- Data is visible in the URL
- Useful for search and filtering
- Usually for reading information

- POST sends the data inside the request body.
- POST is better for registration, login, messages, and sensitive form submission.
- Useful for registration/contact/login
- Usually for sending or changing data

Teaching Focus

GET usually places the submitted data in the URL.

Key teaching points

- Keep examples practical and connected to real websites.
- Pause here and stress the rule, because students often confuse this point.
- Link structure, usability, and user experience.

Form Data Transmission Diagram

- User fills form
- ↓
- Browser packages data
- ↓
- HTTP request sent
- ↓
- Server receives and processes

Input Types Overview

- HTML provides many input types for different purposes.
- Common examples include text, email, password, number, radio, checkbox, date, and file.
- Choosing the correct input type improves user experience and data quality.
- Good form design begins with choosing the right field for the right task.

Teaching Focus

HTML provides many input types for different purposes.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Text-Based Inputs

- type="text" is used for general short text.
- type="email" is used for email addresses.
- type="password" hides what the user types.
- type="number" is used for numeric values such as age, quantity, or score.

- radio
- checkbox
- date
- file

HTML Example

```
<input type="text">
<input type="email">
<input type="password">
<input type="number">
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Labels and Accessibility

- A label explains the purpose of an input field.
- The for attribute of <label> should match the id of the input.
- Clicking the label can place the cursor inside the input field.
- Labels improve usability and accessibility for all users.

- text = general text input
- email = email format support
- password = hidden characters
- number = numeric input

HTML Example

```
<input type="text" name="fullName">
<input type="email" name="email">
<input type="password" name="password">
<input type="number" name="age">
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

name, id, value, and placeholder

- name identifies the field when data is submitted.
- id identifies the element inside the page and connects it to a label.
- value can hold a default or current value.
- placeholder displays a hint inside the field before the user types.

- name = the key sent with submitted data
- id = unique identifier used by labels and CSS/JS
- value = current/default value
- placeholder = temporary hint text

Teaching Focus

name identifies the field when data is submitted.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Radio Buttons

- Radio buttons are used when the user must select only one option.
- All related radio buttons must share the same name value.
- Typical examples include gender selection, payment method, or yes/no choices.
- Radio buttons are best for a small set of mutually exclusive options.

HTML Example

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

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Checkboxes

- Checkboxes allow the user to select more than one option.
- They are suitable for interests, skills, agreement items, and optional choices.
- A checkbox can be checked or unchecked independently.
- This makes checkboxes different from radio buttons.

HTML Example

```
<input type="checkbox" id="html"
name="skills" value="html">
<label for="html">HTML</label>
<input type="checkbox" id="css"
name="skills" value="css">
<label for="css">CSS</label>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Select Menus and Options

- The <select> element creates a drop-down menu.
- The <option> element defines each choice inside the menu.
- Drop-down menus are useful when the number of choices is fixed.
- They save space and keep the form organized.

HTML Example

```
<select name="department">
  <option>Computer Programming</option>
  <option>Electronics</option>
</select>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Textarea for Longer Messages

- A textarea is used when the user needs more space to write.
- Common examples are comments, feedback, requests, and contact messages.
- Unlike a standard input, a textarea can contain multiple lines.
- This makes it suitable for open-ended user communication.

HTML Example

```
<textarea name="message" rows="5"
cols="30"></textarea>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Form Buttons

- submit sends the form data to the destination.
- reset clears the form fields and returns them to their initial state.
- button creates a general button with no special form action by default.
- Choosing the correct button type prevents confusion and errors.

HTML Example

```
<button type="submit">Submit</button>
<button type="reset">Reset</button>
<button type="button">Help</button>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Validation Basics

- Validation helps us collect more accurate and complete data.
- HTML offers useful attributes such as required, min, max, minlength, maxlength, and pattern.
- These tools improve form quality before the data even reaches the server.
- Good validation makes forms clearer and more reliable.

Teaching Focus

Validation helps us collect more accurate and complete data.

Key teaching points

- Keep examples practical and connected to real websites.
- Pause here and stress the rule, because students often confuse this point.
- Link structure, usability, and user experience.

Validation Examples

- required makes a field mandatory.
- min and max set numeric limits.
- minlength and maxlength control the length of text.
- pattern checks whether the input matches a rule.

HTML Example

```
<input type="email" required>
<input type="number" min="1" max="100">
<input type="text" pattern="[A-Za-z]{3,}">
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Building a Better Form

- A better form has a logical order of fields.
- Labels should be clear and close to the related input.
- The correct input type should be chosen for each task.
- The form should be simple enough to use without confusion.

- Use a logical order.
- Use meaningful labels.
- Choose the correct input type.
- Keep the structure simple and readable.

Teaching Focus

A better form has a logical order of fields.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

fieldset and legend (Grouping related fields)

- fieldset groups related form controls together.
- legend gives a title to that group.
- These elements make long forms easier to read.
- They are especially useful when a form has several sections.

HTML Example

```
<fieldset>
  <legend>Contact Information</legend>
  <label for="name">Name</label>
  <input id="name" type="text">
</fieldset>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Example: Contact Form Structure

- A contact form usually includes name, email, subject, message, and a submit button.
- Its goal is clear communication, not data overload.
- A good contact form is short, direct, and easy to complete.
- This type of form appears on business, portfolio, and university websites.

- **A contact form usually includes name, email, subject, message, and a submit button.**

Teaching Focus

```
<form>
  Name
  Email
  Subject
  Message
  Submit
</form>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Example: Registration Form Structure

- A registration form often includes username, email, password, department, interests, and agreement options.
- This type of form usually collects more structured information than a contact form.
- Grouping and validation are especially important in registration forms.
- The layout should remain clear even when more fields are added.

A registration form often includes username, email, password, department, interests, and agreement options.

Teaching Focus

```
<form>
  Username
  Password
  Department
  Interests
  Agreement checkbox
</form>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Common Form Mistakes

- Fields without labels confuse users.
- Missing name attributes prevent data from being submitted correctly.
- Using the wrong input type reduces usability.
- Too many unnecessary fields can frustrate users and reduce form completion.

Fields without labels confuse users.

Teaching Focus

- Missing labels
- Missing name attributes
- Wrong input type
- Too many fields
- Unclear layout

Key teaching points

- Keep examples practical and connected to real websites.
- Pause here and stress the rule, because students often confuse this point.
- Link structure, usability, and user experience.

From Forms to Media

Transition to the second half of the lesson

- Web pages are not only made of text and input fields.
- Modern pages often include images, audio, video, maps, and external content.
- Embedded media makes pages richer and more engaging.
- The key is to use media meaningfully, not excessively.

Key Idea

Web pages are not only made of text and input fields.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Explain how media should support the content, not distract from it.

What Is Embedded Media?

- Embedded media means placing media content directly inside a web page.
- Examples include images, audio, video, maps, and content from other websites.
- Embedded media can improve both communication and user engagement.
- However, media should support the content, not distract from it.

- Audio
- Video
- Maps
- External content inside another page

Key Idea

Embedded media means placing media content directly inside a web page.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Explain how media should support the content, not distract from it.

Image Review

- The <img> element is used to display an image on the page.
- The src attribute gives the image path or URL.
- The alt attribute provides alternative text if the image cannot be seen.
- Width and height can help control image dimensions.

Media Example

```
<img src="campus.jpg" alt="University campus" width="400">
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Explain how media should support the content, not distract from it.

Audio in HTML

- The <audio> element allows us to place sound on a page.
- The controls attribute shows play, pause, and volume controls.
- Audio can be useful for language learning, music samples, podcasts, or pronunciation examples.
- It should be used only when it adds clear value.

Media Example

```
<audio controls>
  <source src="lesson.mp3"
type="audio/mpeg">
</audio>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Explain how media should support the content, not distract from it.

Video in HTML

- The <video> element allows us to display video directly in the browser.
- The controls attribute lets the user play, pause, and control the video.
- Video can be useful for tutorials, product demos, and promotional content.
- As with audio, it should be purposeful and not excessive.

Media Example

```
<video controls width="420">
  <source src="intro.mp4"
type="video/mp4">
</video>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Explain how media should support the content, not distract from it.

iframe

- An iframe displays external content inside the current page.
- It can be used to embed videos, maps, or another web page.
- This makes it possible to show outside content without leaving the site.
- An iframe should be used carefully and only when it improves the page.

Media Example

```
<iframe src="https://example.com"
width="500" height="300"></iframe>
```

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Explain how media should support the content, not distract from it.

Embedding YouTube Videos

- YouTube videos are commonly embedded with an iframe.
- This is useful for lessons, tutorials, presentations, and portfolio pages.
- Embedded video can improve understanding by combining text with demonstration.
- A page should still provide context around the video.

Key Idea

YouTube videos are commonly embedded with an iframe.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Explain how media should support the content, not distract from it.

Embedding Maps

- Embedded maps are common on contact pages and company websites.
- They help users find a location more easily.
- Maps are especially useful for campuses, offices, stores, and event venues.
- This is one of the most practical uses of embedded external content.

Key Idea

Embedded maps are common on contact pages and company websites.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Explain how media should support the content, not distract from it.

Best Practices for Embedded Media

- Use media only when it improves understanding or user experience.
- Keep pages fast by avoiding unnecessarily large files.
- Provide labels, titles, or context for media elements.
- Balance media with text so the page remains clear and readable.

Key Idea

Use media only when it improves understanding or user experience.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Explain how media should support the content, not distract from it.

Common Media Mistakes

- Using very large files can slow down the page.
- Autoplaying audio or video can annoy users.
- Missing controls can make the media hard to use.
- Media without explanation or poor placement can weaken the design.

Teaching Focus

Using very large files can slow down the page.

Key teaching points

- Keep examples practical and connected to real websites.
- Pause here and stress the rule, because students often confuse this point.
- Link structure, usability, and user experience.

Mini Practice Activity

- Create a simple HTML page with one realistic form.
- Add at least one validated input field.
- Include one media element such as an image, audio clip, video, or embedded map.
- Keep the page organized, clear, and purposeful.

Teaching Focus

Create a simple HTML page with one realistic form.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Lesson Summary

- Forms collect user information and support interaction.
- Better forms use labels, suitable input types, validation, and clear organization.
- Embedded media enriches web pages through images, audio, video, and external content.
- Effective web design combines usability, clarity, and meaningful presentation.

Teaching Focus

Forms collect user information and support interaction.

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.

Quick Check / Exit Questions

- What is the difference between GET and POST?
- Why is the name attribute important in forms?
- What is the difference between radio buttons and checkboxes?
- What does required do, and what is an iframe used for?

Teaching Focus

What is the difference between GET and POST?

Key teaching points

- Keep examples practical and connected to real websites.
- Contrast correct usage with common mistakes.
- Link structure, usability, and user experience.