Day 1: Foundations of HTML5 and Document Structure

HTML5 Overview

- HTML5 is the latest iteration of HTML and introduces a range of new elements, attributes, and APIs that make it easier to build modern, feature-rich websites.

- Why HTML5?

- HTML5 was introduced to standardize the web across devices (desktop, mobile).

- It allows web pages to function better offline and integrate multimedia elements (audio, video) without relying on third-party plugins like Flash.

Key Features:

- New semantic elements.

- Native support for audio, video, and canvas graphics.

- Improved accessibility.

- Local storage and offline functionality via HTML5 APIs.

HTML5 Document Structure:

- Doctype: The `<!DOCTYPE html>` declaration tells the browser to use the HTML5 standards mode.

- HTML5 Boilerplate:

Example:

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>HTML5 Example</title>

</head>

<body>

<h1>Welcome to HTML5</h1>

<p>This is a simple HTML5 document.</p>

</body>

</html>

```

Meta Tags and Viewport Settings:

- The `<meta>` tag provides metadata about the HTML document, such as the charset and viewport settings.

- Viewport meta tag is critical for responsive design:

Example:

```html

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

```

HTML5 Semantic Elements:

- Semantic elements introduce a cleaner, more meaningful structure to HTML documents. They replace generic containers like `<div>` with meaningful tags like `<header>`, `<nav>`, and `<article>`.

- Benefits:

- Better for SEO (search engines can better understand your content).

- Improves accessibility for screen readers.

Example:

```html

<header>

<h1>Site Title</h1>

<nav>

<ul>

<li><a href="">Home</a></li>

<li><a href="">About</a></li>

<li><a href="">Contact</a></li>

</ul>

</nav>

</header>

<main>

<section>

<article>

<h2>Article Title</h2>

<p>Article content goes here...</p>

</article>

</section>

</main>

<footer>

<p>&copy; 2024 My Website</p>

</footer>

```

---

Day 2: Text Elements, Media Embedding, and Graphics

Text-level Semantics:

- HTML5 adds new inline elements to describe specific pieces of content.

- `<mark>`: Highlights text.

- `<time>`: Represents time or a date.

- `<progress>`: Displays a progress bar.

- `<meter>`: Shows a scalar measurement.

Examples:

```html

<p>This is a <mark>highlighted</mark> word.</p>

<p>Today is <time datetime="2024-09-25">September 25, 2024</time>.</p>

<progress value="70" max="100">70%</progress>

<meter min="0" max="100" value="50">50</meter>

```

Embedding Multimedia:

- HTML5 allows native embedding of multimedia elements, eliminating the need for external plugins.

- Video:

- The `<video>` element embeds video content directly in the webpage.

- Attributes: `controls`, `autoplay`, `muted`, `loop`, `poster`.

Example:

```html

<video controls>

<source src="video.mp4" type="video/mp4">

Your browser does not support the video tag.

</video>

```

- Audio:

- The `<audio>` element embeds sound files.

- Attributes: `controls`, `autoplay`, `loop`.

Example:

```html

<audio controls>

<source src="audio.mp3" type="audio/mp3">

Your browser does not support the audio element.

</audio>

```

Canvas for Graphics:

- The `<canvas>` element is used to draw graphics via JavaScript. It supports shapes, images, and text but relies on JavaScript for rendering.

Example:

```html

<canvas id="myCanvas" width="200" height="100"></canvas>

<script>

var canvas = document.getElementById('myCanvas');

var context = canvas.getContext('2d');

context.fillStyle = 'red';

context.fillRect(20, 20, 150, 100);

</script>

```

- SVG (Scalable Vector Graphics):

- Unlike Canvas, SVG is used to create resolution-independent vector graphics directly within HTML.

Example:

```html

<svg width="100" height="100">

<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />

</svg>

```

---

Day 3: Forms in HTML5

New HTML5 Input Types:

- HTML5 introduces several new input types for better data input handling:

- `email`: Ensures email format.

- `url`: Ensures valid URL.

- `tel`: Phone number input.

- `date`: Provides a date picker.

- `number`: Restricts to numerical input.

- `range`: Creates a slider control.

Examples:

```html

<form>

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

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

<label for="url">Website:</label>

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

<label for="date">Date of Birth:</label>

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

<label for="quantity">Quantity:</label>

<input type="number" id="quantity" name="quantity" min="1" max="10">

<label for="volume">Volume:</label>

<input type="range" id="volume" name="volume" min="0" max="100">

</form>

```

Form Validation:

- HTML5 provides native form validation attributes like `required`, `pattern`, `min`, `max`, and `maxlength`.

Examples:

```html

<form>

<label for="username">Username:</label>

<input type="text" id="username" name="username" pattern="[A-Za-z]{3,}" required title="Minimum 3 letters">

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

<input type="password" id="password" name="password" minlength="8" required>

</form>

```

Datalist for Autocompletion:

- The `<datalist>` element provides autocompletion for input fields.

Example:

```html

<label for="browser">Choose your browser:</label>

<input list="browsers" id="browser" name="browser">

<datalist id="browsers">

<option value="Chrome">

<option value="Firefox">

<option value="Safari">

<option value="Edge">

</datalist>

```

---

Day 4: HTML5 APIs and JavaScript Integration

Geolocation API:

- The Geolocation API enables you to obtain the user's geographical position. This can be used in mapping services, weather apps, etc.

Example:

```html

<script>

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(showPosition);

}

function showPosition(position) {

document.getElementById("demo").innerHTML =

"Latitude: " + position.coords.latitude +

"<br>Longitude: " + position.coords.longitude;

}

</script>

<p id="demo"></p>

```

Web Storage API:

- HTML5 introduces two storage mechanisms: `localStorage` and `sessionStorage` for client-side data storage.

- localStorage: Data persists even after the browser is closed.

- sessionStorage: Data persists only for the duration of the session.

Examples:

```html

<script>

localStorage.setItem('username', 'JohnDoe');

document.getElementById('username').innerHTML = localStorage.getItem('username');

</script>

<p id="username"></p>

```

---

Day 5: Advanced HTML5 Elements and Final Project

Web Workers:

- Web Workers allow background threads to execute JavaScript code, improving performance without blocking the main UI.

Example:

```html

<script>

var worker = new Worker('worker.js');

worker.onmessage = function(event) {

document.getElementById('result').innerHTML = event.data;

};

worker.postMessage('Start');

</script>

<p id="result"></p>

```

Responsive Images:

- HTML5 allows you to deliver optimized images for different screen sizes using the `srcset` attribute and the `<picture>` element.

Example:

```html

<img src="small.jpg"

srcset="large.jpg 1024w, medium.jpg 640w, small.jpg 320w"

alt="Responsive Image">

```

ARIA for Accessibility:

- ARIA attributes make HTML elements more accessible, improving usability for users who rely on screen readers.

Example:

```html

<button aria-label="Close Menu">X</button>

```

---

This detailed explanation covers a range of topics with examples to help you grasp HTML5 more thoroughly. Let me know if you need additional examples or further clarification on any concepts!