HTML5

Rohan Chhetry

- HTML5 is less strict and more forgiving in terms of syntax.
- One key change is the self-closing syntax for certain elements.

XHTML

```
<img src="image.jpg" alt="Description" />
```

HTML5

```
<img src="image.jpg" alt="Description">
```

XHTML vs HTML5

XHTML (eXtensible HyperText Markup Language)

New Feature of HTML5

01

Semantic Markup

Use of meaningful tags to structure content on a webpage.

```
<header>
  <h1>Page Title</h1>
  <p>Subtitle or tagline goes here</p>
</header>

<article>
  <h2>Article Title</h2>
  <p>Content of the article...</p>
</article>

<aside>
  <h3>Related Content</h3>
  <p>Content related to the main article...</p>
</aside>

<footer>
  <p>Copyright © 2023 Your Name</p>
</footer>
```

Question 01

Create a simple webpage with a header, a main section, and a footer using HTML5 semantic elements

Solution 01

```
<!DOCTYPE html>
<html>
<head>
 <title>HTML5 Semantic Elements</title>
</head>
<body>
 <header>
  <h1>Header</h1>
 </header>
 <main>
  <section>
   <h2>Main Section</h2>
   <p>Content goes here...</p>
  </section>
 </main>
 <footer>
  <p>Footer</p>
 </footer>
</body>
</html>
```

02

Native Audio and Video Support

- HTML5 includes the <audio> and <video> elements
- Allowing developers to embed multimedia content without relying on third-party plugins like Flash

Question 02

Embed a video of your choice using the <video> element.

Solution 02

```
<video width="320" height="240" controls>
  <source src="example.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

03

Canvas Elements

- The <canvas> element enables drawing graphics and animations directly on the web page using JavaScript.
- Useful for creating interactive games, charts, and visualizations.

Question 03

Create a simple HTML page with a canvas element, and use JavaScript to draw a rectangle on the canvas

Solution 03

```
<!DOCTYPE html>
<html>
<head>
 <title>Canvas Example</title>
</head>
<body>
 <canvas id="myCanvas" width="200" height="100" style="border:1px solid #000;">
</canvas>
 <script>
  var canvas = document.getElementById("myCanvas");
  var ctx = canvas.getContext("2d");
  ctx.fillStyle = "#FF0000";
  ctx.fillRect(10, 10, 150, 80);
 </script>
</body>
</html>
```

04

Form Enhancement

- HTML5 introduced new input types such as <input type="date">, <input type="email">, <input type="url">, and more, providing better user experience and validation.
- Additionally, the <datalist> element allows for pre-defined options in dropdown lists.

Question 04

Create a form with input fields for name, email, and a date of birth using HTML5 input type

Solution 04

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

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <label for="dob">Date of Birth:</label>
  <input type="date" id="dob" name="dob" required>

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

05

Local Storage

- HTML5 introduced the **localStorage** and **sessionStorage** APIs, enabling web applications to store data locally on the user's device.
- Providing a more efficient alternative to cookies for client-side storage

Question 05

Use the localStorage API to store and retrieve a user's name on a webpage. Display a personalized greeting.

Solution 05

```
<script>
  // Store data
  localStorage.setItem('username', 'John');

  // Retrieve data
  var username = localStorage.getItem('username');

  // Display personalized greeting
  document.write('Hello, ' + username + '!');
</script>
```

06 Web Workers

- Web Workers allow the execution of JavaScript code in the background, separate from the main thread.
- This enables concurrent processing and improves the performance of web applications, particularly for computationally intensive tasks.

Question 06

Implement a simple web worker that calculates the square of a number in the background

Solution 06

```
<script>
  var worker = new Worker('worker.js');

  worker.onmessage = function(event) {
    console.log('Result: ' + event.data);
  };

  worker.postMessage(5);
</script>
```

worker.js

```
onmessage = function(event) {
  var result = event.data * event.data;
  postMessage(result);
};
```

07

Geolocation API

- Enables websites to access the user's location information with their consent.
- This feature is valuable for location-based services and applications.

Question 07

Use the Geolocation API to display the user's current latitude and longitude.

Solution 07

```
<script>
  navigator.geolocation.getCurrentPosition(function(position) {
    console.log('Latitude: ' + position.coords.latitude);
    console.log('Longitude: ' + position.coords.longitude);
  });
</script>
```

08

Responsive Images

- The <picture> and <source> elements, along with the srcset attribute, allow developers to provide multiple image sources based on the user's device and screen size.
- This contributes to better responsiveness and improved performance

Question 08

Implement an image tag with multiple sources for different screen sizes using the <picture> element.

Solution 08

```
<picture>
  <source media="(min-width: 768px)" srcset="large.jpg">
  <img src="small.jpg" alt="Description">
</picture>
```

09

Drag and Drop

HTML5 introduced native support for drag-and-drop operations, making it easier to implement interactive features like file uploads and rearranging elements on a page

Question 09

Create a webpage with two draggable elements and a drop zone. Use the drag-and-drop API to handle the interaction.

Solution 09

```
<div id="draggable1" draggable="true" ondragstart="event.dataTransfer.setData('text/plain', 'Element 1')">Element 1</div>
<div id="draggable2" draggable="true" ondragstart="event.dataTransfer.setData('text/plain', 'Element 2')">Element 2</div>
<div id="dropZone" ondrop="drop(event)" ondragover="allowDrop(event)">Drop Zone</div>

<script>
 function allowDrop(event) {
  event.preventDefault();
 }

 function drop(event) {
  event.preventDefault();
  var data = event.dataTransfer.getData('text/plain');
  event.target.innerHTML = data;
 }
</script>
```

10

Contenteditable Attribute

The **contenteditable** attribute allows developers to make elements editable, enabling the creation of rich text editors and interactive content.

Question 10

Create a simple webpage with a element that has the contenteditable attribute. Allow users to edit the content.

Solution 10

```
<div contenteditable="true">
  <p>This content can be edited by the user.</p>
</div>
```