

Html5 Assignment

1. What are the new tags added in HTML5?

- HTML5 introduced several new tags that enhance the semantic structure of web documents and improve the overall functionality and accessibility of web pages. Here are some of the key new tags added in HTML5:

1.<article>: Represents a self-contained piece of content that could be distributed independently, such as a news article or a blog post.

2.<aside>: Defines content that is tangentially related to the content around it, often used for sidebars or additional information.

3.<footer>: Represents the footer of a section or page, typically containing information about the author, copyright, or related links.

4.<header>: Defines the introductory content or navigational links for a section or page, often containing headings, logo, or other introductory elements.

5.<main>: Specifies the dominant content of the <body> of a document. There should be only one <main> element per document.

6.<nav>: Defines a section of navigation links.

7.<section>: Represents a thematic grouping of content, typically with a heading, used to divide a document into sections.

8.<summary>: Provides a summary or heading for the content in a <details> element.

9.<time>: Represents a specific period in time, including dates and times.

10.<video>: Embeds video content into a web page, with support for multiple video formats and playback controls.

11.<audio>: Embeds audio content, such as music or sound effects, into a web page, with support for multiple audio formats and playback controls.

13. <svg>: Allows the inclusion of scalable vector graphics directly within HTML.

2. How to embed audio and video in a webpage?

- **Embedding Audio** To embed audio, use the <audio> element. You can include various audio formats and provide fallback options if the user's browser doesn't support a specific format.

Example:-

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

Explanation:

controls attribute adds playback controls (play, pause, volume, etc.).

<source> elements specify different audio file formats to ensure compatibility across browsers.

- **Embedding Video** To embed video, use the <video> element. Similar to audio, you can include various video formats and provide fallback options.

Example:-

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

Explanation:

width and height attributes set the size of the video player.

controls attribute adds playback controls (play, pause, volume, etc.).

<source> elements specify different video file formats.

Additional Options and Attributes

Autoplay: If you want the audio or video to start playing automatically, add the autoplay attribute.

Loop: To make the media play in a loop, use the loop attribute.

Muted: To start the media with sound muted, use the muted attribute.

3. Semantic element in HTML5?

- Semantic elements in HTML5 are elements that clearly describe their meaning in a human- and machine-readable way. They help to define the structure of a webpage and improve accessibility, search engine optimization, and maintainability. Here's a look at some of the key semantic elements introduced or emphasized in HTML5:

- **1.<article>**: Represents a self-contained piece of content that could be distributed independently, such as a news article or a blog post.
- **2.<aside>**: Defines content that is tangentially related to the content around it, often used for sidebars or additional information.
- **3.<footer>**: Represents the footer of a section or page, typically containing information about the author, copyright, or related links.
- **4.<header>**: Defines the introductory content or navigational links for a section or page, often containing headings, logo, or other introductory elements.
- **5.<main>**: Specifies the dominant content of the <body> of a document. There should be only one <main> element per document.
- **6.<nav>**: Defines a section of navigation links.
- **7.<section>**: Represents a thematic grouping of content, typically with a heading, used to divide a document into sections.
- **8.<summary>**: Provides a summary or heading for the content in a <details> element.
- **9.<time>**: Represents a specific period in time, including dates and times.

4. Canvas and SVG tags

- **<canvas> Element**

The <canvas> element is a versatile tool for drawing graphics using JavaScript. It provides a bitmap-based drawing surface where you can create and manipulate images and shapes dynamically.

Example:-

```
<canvas id="myCanvas" width="200" height="100"></canvas>
<script>
  var canvas = document.getElementById('myCanvas');
  var ctx = canvas.getContext('2d');
  ctx.fillStyle = 'blue';
  ctx.fillRect(10, 10, 150, 75);
</script>
```

Key Features

- **Dynamic Drawing:** You use JavaScript to draw and manipulate graphics. The Canvas API provides methods to draw shapes, text, and images.
- **Pixel-based:** Graphics are rendered in a raster format, meaning they are made up of individual pixels.
- **Performance:** Well-suited for high-performance applications like games or real-time visualizations where you need to draw complex scenes quickly.

- **<svg> Element**

The <svg> element is used to define vector-based graphics, which are scalable and resolution-independent. SVG stands for Scalable Vector Graphics and is defined using XML.

Example:-

```
<svg width="200" height="100">  
<rect x="10" y="10" width="150" height="75" fill="blue" />  
</svg>
```

Key Features

- **Vector-based:** Graphics are defined using geometric shapes, so they scale without loss of quality.
- **DOM Integration:** SVG elements are part of the DOM, allowing you to manipulate them with CSS and JavaScript.
- **Declarative Syntax:** You define the graphics directly in HTML or XML.