Module (HTML 5)

1.What are the new tags added in HTML 5?

Ans. HTML5 introduced several new tags to improve web development by enhancing semantics, multimedia capabilities, and form controls.

1. article

2. aside

3. details

4. figcaption

5. figure

6. footer

7. header

8. main

9. mark

10. nav

These elements were introduced to handle multimedia content without the need for external plugins:

1. audio

2. video

3. source

4. track

HTML5 introduced new form elements to improve user interaction and data input:

1. datalist

2. keygen

3. output

These elements provide interactive features to improve user experience:

1. progress

2. meter

3. summary

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

Ans. ===Embedding Audio

You can use the <audio> element to embed audio files

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Audio Embedding Example</title>

</head>

<body>

<h1>Audio Embedding Example</h1>

<audio controls>

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

<source src="audiofile.ogg" type="audio/ogg">

</audio>

</body>

</html>

- The controls attribute adds play, pause, and volume controls.

=== Embedding Video

For video, the <video> element is used.

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Video Embedding Example</title>

</head>

<body>

<h1>Video Embedding Example</h1>

<video controls width="600">

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

<source src="videofile.webm" type="video/webm">

<source src="videofile.ogv" type="video/ogg">

</video>

</body>

</html>

- The controls attribute adds play, pause, and volume controls.

3.Semantic element in HTML5?

Ans. These elements make the HTML more readable and easier to maintain, both for developers and for search engines that parse the content.

1. header: Represents a container for introductory content or a set of navigational links. A header element typically contains one or more heading elements (<h1> to <h6>), a logo, or any other introductory content.

2. nav: Defines a set of navigation links. It is intended for major block of navigation links such as a table of contents or a menu.

3. section: Represents a generic section of a document, typically with a heading. A section element groups content into thematic sections.

4. article: Represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable, such as a blog post or news article.

5. aside: Represents a portion of a document whose content is only indirectly related to the document's main content. It is often used for sidebars.

6. footer: Defines a footer for a document or section, typically containing information about the author, copyright data, links to related documents, etc.

7. main: Specifies the main content of a document. The content inside the main element should be unique to the document and not repeated across pages (e.g., blog post, main article).

8. figure and figcaption: The <figure> element is used to encapsulate a visual representation of some content, such as an image, diagram, or code snippet. The <figcaption> element is used to provide a caption for the figure.

4.Canvas and SVG tags

Ans. The HTML <canvas> and <svg> tags are both used to draw graphics on a web page, but they serve different purposes and have different strengths

<canvas> Tag

- Type: Raster-based (bitmap)

- Usage: Suitable for dynamic, script-driven graphics such as games, animations, and real-time visualizations.

- Performance: Generally faster for drawing operations due to its immediate mode graphics rendering model.

- Drawing Model: Imperative. You draw using JavaScript commands, and once something is drawn, it is part of the bitmap and can't be directly manipulated.

- Interaction: Limited. You need to manually handle interactions like mouse events.

- Complexity: Requires more code for creating and managing graphics.

<svg> Tag

- Type: Vector-based

- Usage: Ideal for static or less frequently updated graphics, such as diagrams, icons, and scalable images.

- Performance: Slower for complex scenes with many elements, but better for static images and those requiring frequent manipulation.

- Drawing Model: Declarative. You define shapes and elements directly in the markup, and these elements remain accessible and manipulable.

- Interaction: Built-in. Elements are part of the DOM, so they automatically handle events like clicks and hovers.

- Complexity: Easier to create and manipulate for scalable graphics.