**Module 5) HTML5**

**1.What are the new tags added in HTML5?**

1. `<header>`: Represents a container for introductory content or a set of navigational links.

2. `<nav>`: Defines a section of navigation links.

3. `<section>`: Represents a standalone section of content, such as a chapter in an article.

4. `<article>`: Defines a self-contained composition that could be distributed or reused independently, such as a blog post or news story.

5. `<aside>`: Represents content that is tangentially related to the main content, like a sidebar or advertising content.

6. `<footer>`: Defines a footer for a section or a page, typically containing information like author details, copyright information, and related links.

7. `<main>`: Specifies the main content of a document, excluding headers, footers, and sidebars.

8. `<figure>` and `<figcaption>`: Used to embed images, diagrams, videos, and other media with captions.

9. `<time>`: Represents a specific period in time or a range of time, typically used to provide timestamps or dates.

10. `<mark>`: Highlights text as if it were marked or highlighted using a marker pen.

11. `<progress>`: Displays the progress of a task, such as the completion of a form or download.

12. `<meter>`: Represents a scalar measurement within a known range, like a gauge.

13. `<details>` and `<summary>`: Used to create a disclosure widget, which allows the user to reveal or hide additional information.

14. `<datalist>`: Provides a list of pre-defined options for an `<input>` element.

15. `<output>`: Displays the result of a calculation or user action.

16. `<canvas>`: Provides a drawing surface for graphics and animations using JavaScript.

17. `<audio>` and `<video>`: Embeds audio and video content with native browser support.

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

**Embedding Audio code :-**

<audio controls>

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

Your browser does not support the audio element.

</audio>

**Embedding video code :-**

<video controls width="640" height="360">

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

Your browser does not support the video element.

</video>

**3. Semantic element in HTML5?**

In HTML5, semantic elements are specific HTML elements that provide meaning to the structure of web content, making it more understandable both for browsers and for developers. These elements help convey the intended purpose of the content within the document structure, improving accessibility, search engine optimization (SEO), and overall maintainability of the code.

* **Here are some examples of semantic elements introduced in HTML5:**

**1. `<header>`: Represents the introductory content or a container for a group of introductory content in a section. Typically includes headings, logos, navigation menus, etc.**

**2. `<nav>`: Defines a section of navigation links, such as a menu, within the document.**

**3. `<main>`: Represents the main content of the document, excluding headers, footers, and sidebars.**

**4. `<article>`: Defines a self-contained composition within a document, such as a blog post, news story, or comment.**

**5. `<section>`: Represents a thematic grouping of content within a document, such as chapters, subsections, or different segments.**

**6. `<aside>`: Represents content that is tangentially related to the main content, often used for sidebars or pull quotes.**

**7. `<figure>`: Represents any content that is referenced from the main content, such as images, illustrations, diagrams, etc.**

**8. `<figcaption>`: Provides a caption or description for the content within a `<figure>` element.**

**9. `<footer>`: Represents the footer of a section or the document, often containing copyright information, contact details, or related links.**

**10. `<details>` and `<summary>`: Used to create interactive disclosure widgets to show or hide additional information.**

**4.Canvas and SVG tags?**

Both Canvas and SVG (Scalable Vector Graphics) are HTML elements used for creating graphics and visual content on web pages, but they have different underlying mechanisms and use cases.

1. \*\*Canvas:\*\*

The `<canvas>` element provides a way to draw graphics on a web page using JavaScript. It is a bitmap-based approach, which means that you directly manipulate pixels on the canvas to create your graphics. Here are some key points about the `<canvas>` element:

- Drawing: You use JavaScript to draw shapes, lines, text, images, and more directly onto the canvas. This gives you fine-grained control over individual pixels.

- Performance: Well-suited for complex and interactive graphics, animations, and games.

- Resolution: The canvas has a fixed resolution. It does not scale well to different screen sizes without additional code.

- Animation: Requires continuous redrawing to create animations.

- Examples: Graphing applications, interactive games, complex data visualizations.

Example usage:

```html

<canvas id="myCanvas" width="500" height="300"></canvas>

<script>

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

const ctx = canvas.getContext('2d');

ctx.fillStyle = 'red';

ctx.fillRect(50, 50, 100, 100);

</script>

```

2. \*\*SVG:\*\*

SVG is a markup language for describing two-dimensional vector graphics. It uses XML syntax to define shapes, lines, curves, and other graphical elements. Unlike canvas, SVG is resolution-independent and defines shapes mathematically, making it great for responsive designs and static images. Some important points about SVG:

- Drawing: Graphics are defined using XML-based markup directly in the HTML.

- Scalability: Graphics remain crisp at any size and can easily adapt to different screen sizes and resolutions.

- Interaction: Supports interactivity through CSS and JavaScript, but interactions may require more code than canvas.

- Animation: Allows animation through CSS and SMIL (Synchronized Multimedia Integration Language).

- Examples: Icons, logos, data visualizations, maps.

Example usage:

```html

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

<circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />

</svg>