**Name: Pratham R. Sachdev**

**Assignment – Module 3 (HTML5)**

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

**-**

HTML5 introduced several new tags to enhance semantics, multimedia, and form handling. Some of the key tags include:

**1. Semantic Tags:**

* <article>: Defines independent, self-contained content.
* <section>: Groups related content together.
* <nav>: Defines navigation links.
* <header>: Represents the introductory content or a set of navigational links.
* <footer>: Represents the footer of a section or page.
* <aside>: Represents content related to the surrounding content (like sidebars).
* <main>: Indicates the main content of the document.

**2. Multimedia Tags:**

* <audio>: Embeds audio content.
* <video>: Embeds video content.
* <source>: Specifies multiple media resources for media elements (<audio>, <video>).
* <track>: Specifies text tracks (e.g., captions) for media elements.

**3. Form-Related Tags:**

* <datalist>: Provides an autocomplete feature for form inputs.
* <output>: Represents the result of a calculation or user action.
* <progress>: Displays the progress of a task.
* <meter>: Represents a scalar measurement within a known range (e.g., disk usage).

**4. Other Tags:**

* <canvas>: Provides a drawing surface for graphics via JavaScript.
* <figure>: Represents self-contained content, like illustrations, diagrams, or photos.
* <figcaption>: Provides a caption for the <figure> element.
* <mark>: Highlights text.
* <time>: Represents a specific time or duration.

These additions improve the structure and multimedia capabilities of web pages.

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

**-**

**1. Embedding Audio in HTML5:**

The <audio> tag is used to embed audio files, supporting formats like MP3, Ogg, and WAV.

**Example:**

*<audio controls>*

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

*<source src="audio-file.ogg" type="audio/ogg">*

*</audio>*

**Key Attributes:**

* **controls:** Adds play, pause, and volume control options for the user.
* **autoplay:** Automatically plays the audio when the page loads (without controls, this can be intrusive).
* **loop:** Repeats the audio automatically.
* **preload:** Specifies if and how the browser should load the audio file before playback. Options include:

1. **none:** Do not preload the file.
2. **metadata**: Preload only metadata.
3. **auto**: Preload the entire file.

**2. Embedding Video in HTML5:**

The <video> tag is used to embed video files. It supports formats like MP4, WebM, and Ogg.

**Example:**

*<video width="600" height="400" controls>*

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

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

*</video>*

**Key Attributes:**

* **width** and **height**: Set the dimensions of the video player.
* **controls**: Adds play, pause, volume, and fullscreen controls.
* **autoplay**: Automatically plays the video when the page loads.
* **loop**: Repeats the video when it ends.
* **muted**: Starts the video without sound (commonly used with autoplay).
* **poster**: Specifies an image to display before the video starts playing.

**3.** **Semantic element in HTML5?**

**-** Semantic elements in HTML5 provide meaning to the content, making it easier for browsers, developers, and search engines to understand the structure of a webpage.

**Key Semantic Elements:**

1. **<header>**: Defines the introductory section or navigation links.
2. **<nav>**: Contains navigation links.
3. **<section>**: Groups related content.
4. **<article>**: Represents self-contained content, like a blog post.
5. **<aside>**: Contains related content, such as sidebars.
6. **<footer>**: Defines the footer for a section or page.
7. **<main>**: Denotes the main content of a page.
8. **<figure>**: Groups media content with a caption.
9. **<figcaption>**: Provides a caption for the <figure> element.

These elements improve accessibility and SEO by providing clear content structure.

**4.** **Canvas and SVG tag?**

**-**

**1. <canvas> Tag:**

The <canvas> tag in HTML5 provides a drawing surface for rendering graphics dynamically via JavaScript. It's often used for games, data visualizations, and interactive content.

**Key Features:**

* **Drawing:** You can draw shapes, text, images, and animations using JavaScript.
* **No built-in graphics:** The canvas starts as a blank area, and all rendering is done programmatically.
* **2D/3D support**: Can render in 2D (via getContext('2d')) and also supports WebGL for 3D graphics.

**Example**:

*<canvas id="myCanvas" width="500" height="400"> </canvas>*

*<script>*

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

*var ctx = canvas. getContext('2d');*

*ctx. fillStyle = 'blue';*

*ctx.fillRect(10, 10, 150, 100);*

*</script>*

**2. <svg> Tag:**

The <svg> tag stands for **Scalable Vector Graphics**. It is used to create vector-based graphics directly in the HTML markup. SVG is based on XML and allows for complex images and shapes that are scalable without losing quality.

**Key Features:**

* **XML-based**: You can define shapes, lines, paths, and text using markup.
* **Scalability**: Graphics do not lose quality when resized.
* **Static or interactive**: SVG graphics can be styled and manipulated via CSS and JavaScript.
* **Accessibility**: SVG elements are part of the DOM and can be directly accessed and manipulated.

**Example:**

*<svg width="400" height="200">*

*<circle cx="100" cy="100" r="50" stroke="black" stroke-width="3" fill="red" /> </svg>*

**Differences:**

* **Canvas** is pixel-based and relies on JavaScript for drawing, suitable for dynamic content.
* **SVG** is vector-based, resolution-independent, and great for static or scalable graphics.