**Module (HTML5) – 3**

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

**Ans:-**

* HTML5 introduced several new elements to enhance the structure, semantics, and functionality of web pages. Some of the key new tags include:
* **< section > :** Defines a generic section of a document, used for thematic grouping of content.
* **< header > :** Represents introductory content or a set of navigational links.
* **< footer > :** Represents the footer of a section or page, typically containing metadata, links, or copyright information.
* **< nav > :** Represents a section of the document intended for navigation links.
* **< mark > :** Highlights text for reference or emphasis.
* **< figure > :** Encapsulates media content like images, charts, or diagrams, along with their captions.
* **< aside > :** Represents content indirectly related to the main content, often used for sidebars or pull quotes.
* **< figcaption > :** Provides a caption or explanation for the content inside a <figure> element.
* **< time > :** Represents a specific time or date.
* **< progress > :** Displays the progress of a task, such as downloading or uploading.
* **< main > :** Specifies the main content of a document, central to the document's purpose.

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

**Ans:-**

* HTML5 makes it easy to embed audio and video content directly into webpages using the <audio> and <video> elements.
* **Embedding Audio:**
* The controls attribute adds play, pause, and volume controls.
* Multiple <source> elements can be used to provide different audio formats for better browser compatibility.
* **Embedding Video:**
* The controls attribute adds playback controls like play, pause, and volume.
* Multiple <source> elements allow for different video formats, ensuring wider compatibility across browsers.

1. **Semantic element in HTML5?**

**Ans:-**

* Semantic elements in HTML5 provide meaning to the structure of a webpage, making it easier for both developers and browsers to understand the content. Some key semantic elements include:
* **< article > :** For independent, self-contained content that makes sense on its own.
* **< aside > :** For content related to the main content but not central to it, often used for sidebars.
* **< footer > :** For footer content of a page or section.
* **< header > :** For introductory content or navigation links at the beginning of a section.
* **< main > :** For the main content of the document, central to its purpose.
* **< nav > :** For groups of navigation links.
* **< section > :** For grouping related content within a document.
* These elements improve accessibility and SEO by providing more meaningful HTML structure.

1. **Canvas and SVG tags.**

**Ans:-**

* **<canvas> Element:**
* The <canvas> element is used for drawing graphics on the fly using JavaScript. It provides a pixel-based, bitmap surface that is resolution-dependent.
* The <canvas> element itself is blank; all drawing is done via JavaScript.
* **<svg> Element:**
* The <svg> (Scalable Vector Graphics) element is used for creating vector-based graphics that are scalable and resolution-independent.
* SVG graphics can be styled with CSS and manipulated with JavaScript.
* **Difference Between <canvas> and <svg>:**
* <canvas> is better for real-time graphics, games, and dynamic images created with JavaScript.
* <svg> is ideal for static, vector-based images that require scaling and high-quality rendering.
* Both <canvas> and <svg> have their own strengths and use cases, depending on the needs of your project.