**MODULE – 3**

**HTML 5**

**1) What are the new TAGS of HTML5 ?**

**ANS.** HTML5 introduced several new semantically meaningful tags.

|  |  |
| --- | --- |
| **Tags (Elements)** | **Description** |
| <article> | Represents an independent piece of content of a document, such as a blog entry or newspaper article |
| <aside > | Represents a piece of content that is only slightly related to the rest of the page. |
| <audio> | Defines an audio file. |
| <canvas> | This is used for rendering dynamic bitmap graphics on the fly, such as graphs or games. |
| <command> | Represents a command the user can invoke. |
| <datalist> | Together with the a new list attribute for input can be used to make comboboxes |
| <details> | Represents additional information or controls which the user can obtain on demand |
| <embed> | Defines external interactive content or plugin. |
| <figure> | Represents a piece of self-contained flow content, typically referenced as a single unit from the main flow of the document. |
| <footer> | Represents a footer for a section and can contain information about the author, copyright information, et cetera. |
| <header> | Represents a group of introductory or navigational aids. |
| <hgroup> | Represents the header of a section. |
| <keygen> | Represents control for key pair generation. |
| <mark> | Represents a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context. |
| <meter> | Represents a measurement, such as disk usage. |
| <nav> | Represents a section of the document intended for navigation. |
| <output> | Represents some type of output, such as from a calculation done through scripting. |
| <progress> | Represents a completion of a task, such as downloading or when performing a series of expensive operations. |
| <ruby> | Together with <rt> and <rp> allow for marking up ruby annotations. |
| <section> | Represents a generic document or application section |
| <time> | Represents a date and/or time. |
| <video> | Defines a video file. |
| <wbr> | Represents a line break opportunity. |

**2) How to embed Audio and Video in Webpage ?  
ANS.** Save your audio and video files in a new directory on your computer. Create a new HTML file in the same directory, called index. html . Add <audio> and <video> elements to the page; make them display the default browser controls.

Embedding Audio in a Webpage

**The <audio> Tag:**

The <audio> element is used to embed sound content, such as music or podcasts, into a webpage. It can play multiple formats such as MP3, WAV, or OGG.

**Attributes of the <audio> Tag:**

**src:** Specifies the URL of the audio file.

**controls:** Adds play, pause, and volume controls for the user.

**autoplay:** Starts playing the audio automatically when the page loads.

**loop:** Makes the audio play in a continuous loop.

**muted:** Mutes the audio by default when it loads.

**preload:** This attribute specifies if the browser should preload the entire audio file or just a portion of it (values can be "auto," "metadata," or "none").

Multiple Sources for Compatibility:

Since different browsers support different audio formats, it is a good practice to provide multiple audio sources using <source> elements within the <audio> tag. The browser will choose the first format it supports.

**Fallback Content:**

You can provide fallback content between the opening and closing <audio> tags for browsers that do not support the audio element. This might be a message or a link to download the audio file.

Embedding Video in a Webpage

**The <video> Tag:**

The <video> element is used to embed video content into a webpage. It supports formats like MP4, WebM, and OGG, and can handle various types of video content like movies, tutorials, or advertisements.

Attributes of the <video> Tag:

src: Specifies the URL of the video file.

controls: Adds controls for play, pause, volume, and fullscreen toggle.

width and height: Define the dimensions of the video player.

autoplay:

**3) Semantic Elements in HTML5 ?**

**ANS.** Semantic HTML elements are those that clearly describe their meaning in a human- and machine-readable way. Elements such as <header> , <footer> and <article> are all considered semantic because they accurately describe the purpose of the element and the type of content that is inside them.

**4) Canvas and SVG Tags**

**ANS.** The canvas and SVG tags are used for different purposes in web pages, and have different characteristics:

**CANVAS**

This tag is used to create complex, interactive graphics, like games, that require dynamic visual effects. It's based on pixels, so it's ideal for small surface areas or a large number of objects. However, once a shape is drawn, it can't be changed unless the entire scene is redrawn.

**SVG**

This tag is used to embed vector-based graphics in web pages. It's based on mathematical metadata, so it's ideal for large surface areas or a small number of objects. SVG graphics don't lose quality when scaled, making them ideal for responsive design. SVGs are also text-based, so they can be made accessible to screen readers.

**Here are some other differences between the canvas and SVG tags:**

**FLEXIBILITY**

Canvas is a lower level API than SVG, so it offers more flexibility in exchange for greater complexity.

**JAVASCRIPT**

Canvas requires JavaScript to be added after the rectangular canvas area is created. SVG graphics can have JavaScript event handlers attached to them.

**BROWSER RENDERING**

The browser automatically re-renders an SVG shape if its attributes are changed. Once a graphic is drawn on the canvas, the browser forgets it, so the entire scene needs to be redrawn to change its position.