**Module (HTML5) – 3**

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

**Ans**.

New tags introduced by HTML5 improve the semantic structure of web content. Here are a some of the significant new tags along with their justifications and illustrations:

1. "header": Describes a document's or a section's header. Usually, it has the site's logo, navigation menu, or an introduction.

**Example:**

html

<header>

<h1>My Website</h1>

<nav>

<a href="#">Home</a>

<a href="#">About</a>

<a href="#">Contact</a>

</nav>

</header>

2. \<nav>: Represents a section containing navigation links. It is typically placed within the header or footer of a document.

Example:

html

<nav>

<a href="#">Home</a>

<a href="#">About</a>

<a href="#">Contact</a>

</nav>

3. \<section>: Defines a standalone section within a document. It represents a logical grouping of content, such as chapters, tabbed content, or articles.

Example:

html

<section>

<h2>About Us</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

</section>

4. \<article>: Represents a self-contained composition that can be independently distributed or reused, such as a blog post, news article, or forum post.

Example:

html

<article>

<h2>Blog Post Title</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

</article>

5. \<aside>: Defines a section of content that is tangentially related to the main content. It is often used for sidebars, pull quotes, or advertising sections.

Example:

html

<aside>

<h3>Advertisement</h3>

<p>Check out our new product!</p>

</aside>

6. \<footer>: Represents the footer section of a document or a section. It typically contains information about the author, copyright details, or links to related documents.

Example:

html

<footer>

<p>&copy; 2023 My Website. All rights reserved.</p>

</footer>

These are just a few examples of the new tags introduced in HTML5. They help provide clearer semantics to the structure of web documents and improve accessibility and SEO.

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

**Ans.**

The HTML5 "audio" and "video" elements can be used to embed audio and video in webpages. These tags let you indicate the audio or video file's origin and give different properties to govern playback and presentation. Here's an illustration of how to include music and video into a website:

Embedding Audio:

Html

<audio controls>

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

Your browser does not support the audio element.

</audio>

- The audio material is defined by the 'audio' element.

- A simple set of playback controls are added to the audio player using the 'controls' element.

- The'src' property of the'source>' tag, which is nested inside of the 'audio' tag, indicates the location of the audio file's source. To ensure cross-browser compatibility, you can use numerous "source" tags with various file types.

- For browsers that do not support the 'audio>' tag, the message "Your browser does not support the audio element." is shown as a fallback.

Embedding Video:

Html

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

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

Your browser does not support the video element.

</video>

In the above example:

- The video material is defined by the 'video>' element.

- The 'controls' feature provides the video player with a rudimentary set of playback controls.

- The dimensions of the video player are specified by the 'width' and 'height' properties.

- The'source>' tag is nested within the 'video>' tag and uses the'src' property to provide the source of the video file. Again, multiple'source>' tags with different file formats can be provided for greater browser support.

- For browsers that do not support the 'video>' tag, the phrase "Your browser does not support the video element." acts as a fallback message.

Additional attributes and CSS style may be used to modify the appearance and behaviour of the audio and video players. The examples supplied are a good place to start when it comes to embedding audio and video content in your webpages.

1. **Semantic element in HTML5?**

**Ans**

In HTML5, semantic elements are tags that offer meaningful structure and define the material contained within them. They improve a web page's accessibility, search engine optimisation (SEO), and general structure. Browsers, screen readers, and other technologies use semantic components to comprehend the purpose and hierarchy of distinct portions of a webpage.

Here are some examples of semantic elements in HTML5:

1. `<header>`: Represents the introductory or navigational section of a webpage.

Html

<header>

<h1>Welcome to My Website</h1>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Services</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</header>

1. `<nav>`: Defines a section containing navigation links.

Html

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">Products</a></li>

<li><a href="#">Services</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

1. `<main>`: Represents the main content area of a webpage.

Html

<main>

<h1>About Us</h1>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. ...</p>

</main>

1. `<article>`: Represents a self-contained composition that could be independently distributed or syndicated.

Html

<article>

<h2>Blog Post Title</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. ...</p>

</article>

1. `<section>`: Defines a thematic grouping of content within a document.

Html

<section>

<h2>Our Team</h2>

<div class="team-member">

<h3>John Doe</h3>

<p>Position: Web Developer</p>

</div>

<div class="team-member">

<h3>Jane Smith</h3>

<p>Position: Graphic Designer</p>

</div>

</section>

These are just a few examples of semantic elements in HTML5. Using these elements appropriately can improve the structure and accessibility of your web pages, making them easier to understand for both humans and machines.

1. **Canvas and SVG tags**

**Ans.**

Certainly! Both the `<canvas>` and `<svg>` tags are commonly used in web development to create and manipulate graphics, but they have different approaches and purposes. Here's an explanation of each tag along with an example:

1. **`<canvas>` tag:**

The `<canvas>` tag is an HTML element that provides a drawing area on which you can use CSS to render graphics dynamically.The content within the `<canvas>` tag is not directly accessible to screen readers or search engines, as it requires CSS for rendering.

Example:

Html

1)

<canvas id="myCanvas" width="200" height="200"></canvas>

2)

<canvas id="myCanvas" width="400" height="200"></canvas>

<script>

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

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

</script>

In this example, we create a canvas element with an id of "myCanvas" and a width and height of 200 pixels each.

1. **`<svg>` tag:**

The `<svg>` tag is an XML-based markup language for describing two-dimensional vector graphics. It allows you to create scalable vector graphics that can be styled and manipulated using CSS. The content within the `<svg>` tag is accessible to screen readers and search engines since it is defined in XML format.

Example:

Html

1)

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

<circle cx="100" cy="100" r="50" fill="red" />

</svg>

2)

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

<rect x="50" y="50" width="100" height="100" fill="red" />

<circle cx="200" cy="100" r="50" fill="blue" />

</svg>

In this example, we create an SVG element with a width and height of 200 pixels each. Within the SVG, we define a circle element with a center at (100, 100), a radius of 50 pixels, and a fill color of red. SVG elements are self-contained and can be styled or animated using CSS.