**SEMANTICS**

**Assignment#**

**COURSE- FULL STACK WEB DEVELOPMENT**

**Logo

Description automatically generated with low confidence**

**SUBMITTED BY:**

**SAMIM MONDAL**

**SUBMITTED TO:**

**SANKET SINGH**

**1. What are the new things introduced in HTML5?**

**Ans:** Here are some of the key new things introduced in HTML5:

1. **Semantic Elements**: HTML5 introduced several new semantic elements that provide better structure and meaning to web documents. These include **<header>**, **<nav>**, **<article>**, **<section>**, **<aside>**, and **<footer>**, among others. These elements help both developers and search engines understand the content's structure and purpose.
2. **Audio and Video Elements**: HTML5 introduced **<audio>** and **<video>** elements that allow embedding audio and video content directly in web pages, without relying on third-party plugins like Flash. This made multimedia integration much easier and more standardized.
3. **Canvas Element**: The **<canvas>** element introduced a drawing surface for creating graphics and animations using JavaScript. This allowed for dynamic and interactive content, such as games and data visualizations, without relying on external technologies.
4. **SVG (Scalable Vector Graphics)**: HTML5 brought improved support for SVG, allowing vector graphics to be directly embedded in web pages. This enabled high-quality graphics that can be scaled without loss of quality.
5. **Web Forms 2.0**: HTML5 enhanced form controls and introduced new input types, such as **<input type="date">**, **<input type="email">**, **<input type="url">**, and more. Additionally, attributes like **required** and **pattern** provided better validation options.
6. **Web Storage**: HTML5 introduced the **localStorage** and **sessionStorage** APIs, which allow web applications to store data locally on the user's device, providing a way to maintain state and cache data even after the browser is closed.
7. **Web Workers**: HTML5 introduced Web Workers, which enable multi-threading in web applications. This allows for background tasks to be performed without blocking the main user interface thread, improving performance and responsiveness.
8. **Geolocation API**: HTML5 introduced the Geolocation API, which allows web applications to retrieve the user's geographical location. This enabled location-based services and content customization.
9. **Drag and Drop**: HTML5 introduced native drag-and-drop functionality, making it easier to implement interactive interfaces and data manipulation through simple HTML attributes and JavaScript.
10. **Offline Web Applications**: HTML5 introduced the concept of offline web applications using the Application Cache (AppCache). This allowed websites to be cached locally and used even when the user is offline.
11. **WebSockets**: HTML5 introduced the WebSocket API, enabling real-time, full-duplex communication between a client and a server over a single, long-lived connection.
12. **New APIs and Features**: HTML5 brought a plethora of new APIs and features, such as the History API for manipulating the browser history, the File API for working with files, the WebRTC API for real-time communication, and the Web Audio API for advanced audio processing, among others.

**2.** **What are semantic tags? Give couple of examples?**

**Ans:** Semantic tags, often referred to as semantic HTML tags or semantic elements, are HTML elements that convey meaning about the structure and content of a web page to both browsers and developers. They go beyond just defining how content should be displayed and provide additional context to improve accessibility, search engine optimization (SEO), and overall understanding of the document's purpose.

Using semantic tags can enhance the organization and clarity of your HTML code, making it easier for both humans and machines to interpret the content. Here are a couple of examples of semantic tags:

1. **<header>**: The **<header>** tag is used to define the header section of a webpage or a section within the page. It typically contains elements like headings, navigation menus, and other introductory content.

Example:

<header>

<h1>Welcome to Our Website</h1>

<nav>

<ul>

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

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

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

</ul>

</nav>

</header>

**<article>**: The **<article>** tag is used to define a self-contained, independent piece of content within a webpage. It could represent a blog post, news article, forum post, or any other content that can stand alone.

Example:

<article>

<h2>Exploring National Parks</h2>

<p>Discover the beauty of national parks around the world...</p>

<p>Whether you're a hiker, nature lover, or photographer...</p>

</article>

**<section>**: The **<section>** tag is used to group related content together within a webpage. It helps organize content into different thematic sections, making it easier to navigate and understand.

Example:

<section>

<h2>Our Services</h2>

<p>We offer a range of services to meet your needs...</p>

<ul>

<li>Web Design</li>

<li>Graphic Design</li>

<li>SEO Optimization</li>

</ul>

</section>

**<footer>**: The **<footer>** tag is used to define the footer section of a webpage. It often contains information like copyright notices, contact details, and links to related pages or resources.

Example:

<footer>

<p>&copy; 2023 Your Company. All rights reserved.</p>

<p>Contact us at: <a href="mailto:info@example.com">info@example.com</a></p>

</footer>

**3.** **List some uses of semantic tags?**

1. **Ans: Header and Footer Sections:**
   * **<header>**: Represents the header of a document or a section, typically containing site logos, navigation menus, and introductory content.
   * **<footer>**: Represents the footer of a document or a section, often containing copyright information, contact details, and links to related resources.
2. **Navigation:**
   * **<nav>**: Indicates a section of navigation links, like a menu or a navigation bar.
3. **Article Structure:**
   * **<article>**: Defines a self-contained composition that can be distributed or reused independently, like a blog post, news article, or forum post.
4. **Sections and Divisions:**
   * **<section>**: Represents a thematic grouping of content, often with its own heading. It helps organize content into meaningful chunks.
   * **<div>**: Although not semantic by itself, it's often used for layout purposes. Adding class names or roles to **<div>** can add semantic meaning.
5. **Main Content Area:**
   * **<main>**: Identifies the main content area of the document, excluding headers, footers, and sidebars.
6. **Aside Content:**
   * **<aside>**: Represents content that is tangentially related to the content around it, like sidebars, pull quotes, or advertisements.
7. **Figures and Captions:**
   * **<figure>**: Represents any content that is referenced from the main content, such as images, illustrations, diagrams, code snippets, etc.
   * **<figcaption>**: Provides a caption or description for the content within a **<figure>** element.
8. **Lists:**
   * **<ul>**: Defines an unordered list.
   * **<ol>**: Defines an ordered list.
   * **<li>**: Represents a list item within **<ul>** or **<ol>**.
9. **Definition Lists:**
   * **<dl>**: Defines a description list.
   * **<dt>**: Represents a term or name in a description list.
   * **<dd>**: Represents the description or definition of a term in a description list.
10. **Tables:**
    * **<table>**: Represents a data table.
    * **<thead>**, **<tbody>**, **<tfoot>**: Sections within a **<table>** for organizing table headers, body content, and footer content, respectively.
11. **Forms:**
    * **<form>**: Represents an HTML form for user input.
    * **<input>**, **<textarea>**, **<select>**: Various form input elements.
    * **<label>**: Provides a label for form controls.

**THANK YOU!!**