SEMANTICS ASSIGNMENT

1. What are the new things introduced in HML5?

HTML5, the fifth version of the Hypertext Markup Language, brought several new features and improvements to web development. Some of the key additions introduced in HTML5 include:

1. New Semantic Elements:

HTML5 introduced semantic elements like <header>, <nav>, <article>, <section>, <aside>, and <footer>. These elements provide a more meaningful structure to web documents, making it easier for developers to convey the intended meaning of different parts of a page.

1. Audio and Video Support:

HTML5 includes native support for embedding audio and video content without the need for third-party plugins. The <audio> and <video> elements allow developers to integrate multimedia content directly into their web pages.

1. Canvas Element:

The <canvas> element provides a drawing surface for graphics and animations. JavaScript can be used to manipulate and draw on the canvas, enabling the creation of dynamic visual content.

1. New Form Elements:

HTML5 introduced several new form elements, such as <input> types like email, url, number, date, and range. These new input types make it easier to capture specific types of user input and provide better user experiences.

1. Local Storage:

The localStorage and sessionStorage APIs were introduced to enable web applications to store data locally on the user's device. This allows for persisting data across sessions without the need for server-side storage.

1. Web Storage:

The Web Storage API provides a simple key-value store for storing data on the client side. It includes localStorage and sessionStorage for persistent and session-specific storage, respectively.

1. Web Workers:

Web Workers allow developers to run scripts in the background, independent of the main page's user interface. This helps in improving performance and responsiveness, especially for computationally intensive tasks.

1. Geolocation API:

HTML5 introduced the Geolocation API, allowing web applications to access the user's location information. This can be useful for location-aware services and features.

1. Responsive Images:

The <picture> element and the srcset attribute were introduced to facilitate the delivery of responsive images, ensuring that different image resources can be loaded based on the user's device and screen size.

1. Drag and Drop:

HTML5 added native support for drag-and-drop interactions, making it easier to implement intuitive user interfaces for tasks such as file uploads.

1. WebSocket:

The WebSocket API enables bidirectional communication between the web browser and the server, providing a more efficient alternative to traditional HTTP polling.

1. What are semantic tags? Give a couple of examples.

Semantic tags in HTML are elements that carry meaning about the structure and content of the document, rather than just presenting its appearance. Semantic tags provide a clear and descriptive structure to HTML documents, making it easier for both developers and browsers to understand the purpose of different parts of the content. Here are a couple of examples of semantic tags introduced in HTML5:

1. `<header>`:

The <header> element represents a header section typically containing introductory content, navigation links, or other headings. It is used to define the header of a document or a section within a document.

<header>

<h1>Website Title</h1>

<nav>

<ul>

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

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

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

</ul>

</nav>

</header>

1. `<article>``:

The <article> element represents a self-contained piece of content that can be distributed and reused independently. It is often used for blog posts, news articles, forum posts, or other similar content.

<article>

<h2>Article Title</h2>

<p>Content of the article...</p>

<footer>

<p>Published on <time datetime="2024-01-03">January 3, 2024</time></p>

</footer>

</article>

1. <section>:

The <section> element is used to group related content together within a document. It helps in organizing the content and creating a more logical structure.

<section>

<h2>Section Title</h2>

<p>Content of the section...</p>

</section>

1. <nav>:

The <nav> element represents a navigation section containing links to other pages or parts of the current page. It is typically used for menus, navigation bars, or any set of links that guide the user.

<nav>

<ul>

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

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

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

  </ul>

</nav>

1. List some uses of semantic tags.

Semantic tags in HTML provide meaningful information about the structure and content of a document. Here are some common uses of semantic tags:

1. Improved Accessibility:

Semantic tags contribute to better accessibility by providing more information about the purpose of different sections. This helps screen readers and other assistive technologies understand and present content more accurately to users with disabilities.

1. Search Engine Optimization (SEO):

Search engines use semantic tags to better understand the content and context of a webpage. Properly structured content with semantic tags can positively impact search engine rankings.

1. Clear Document Structure:

Semantic tags enhance the overall structure and readability of HTML documents. They make it clear which parts of the content are headers, navigation, main content, articles, sections, etc.

1. Easier Maintenance:

Semantic tags make the code more self-explanatory and easier to maintain. Developers can quickly grasp the structure of a document, making it easier to update or modify the content.

1. CSS Styling and Layout:

Semantic tags can be targeted for styling with CSS, making it easier to create consistent and visually appealing designs. For example, styling a <header> or <nav> element separately from other parts of the document.

1. Responsive Design:

Semantic tags contribute to building responsive web designs. Elements like <header>, <nav>, and <footer> can be styled differently for various screen sizes, providing a better user experience on different devices.

1. Dynamic Content Manipulation:

JavaScript and other scripting languages can leverage semantic tags to manipulate content dynamically. For instance, targeting specific <article> elements for content updates or interactions.

1. Print Styles:

When creating print stylesheets, semantic tags help define how different parts of the document should appear on paper. For example, styling <header> and <footer> for print-specific layouts.

1. Accessibility Landmarks:

Semantic tags such as <header>, <nav>, <main>, <article>, and <footer> act as landmarks for accessibility purposes, allowing users to navigate more efficiently through the content.

1. Standardized Document Outlines:

Semantic tags contribute to creating standardized document outlines. Browsers use these outlines to provide users with an overview of the page's structure, helping with navigation and comprehension.