**Q.1) <!DOCTYPE html> is it a tag of html? If not, what is it and why do we use it?**

**Ans:**

<!DOCTYPE html> is not a tag in HTML; it is called a document type declaration, or DOCTYPE declaration for short. It is used at the beginning of an HTML document to specify the version of HTML or XML that the document adheres to.

The DOCTYPE declaration informs the web browser or other user agents about the markup language and version used in the document. It helps the browser understand how to interpret and render the document correctly. Different versions of HTML have different rules and syntax, so the DOCTYPE declaration ensures that the browser handles the HTML document according to the specified standards.

In the case of <!DOCTYPE html>, it represents the DOCTYPE declaration for HTML5, the latest version of the HTML specification. HTML5 is widely supported by modern web browsers and is backward-compatible with older versions of HTML. By including <!DOCTYPE html> at the beginning of an HTML file, you indicate that the document should be interpreted as HTML5.

Including the DOCTYPE declaration is considered good practice because it ensures consistent rendering across different browsers and devices. It helps avoid compatibility issues and ensures that the HTML document is interpreted correctly by conforming to the appropriate standards.

**Q.2) Explain Semantic tags in html? And why do we need it?**

**Ans.:**

Semantic tags in HTML are specific elements that provide meaning and context to the content they surround. Unlike generic tags like “<div>” and “<span>”, which have no inherent meaning, semantic tags describe the purpose and structure of the content they enclose.

Semantic tags serve several important purposes:

1. Accessibility: Semantic tags help improve web accessibility by providing meaningful information to assistive technologies like screen readers. These technologies can better understand and convey the structure and purpose of the content to users with disabilities.

2. Search engine optimization (SEO): Search engines use semantic tags to understand the content and context of a web page. Using semantic tags can help search engines better index and rank your content, leading to improved visibility and search engine optimization.

3. Code readability and maintainability: Semantic tags make your HTML code more readable and maintainable. By using tags like “<header>”, “<nav>”, “<article>”, “<section>”, and “<footer>”, you provide a clear and structured hierarchy to your content, making it easier for developers to understand and maintain the codebase.

Some commonly used semantic tags and their purposes are:

- “<header>”: Represents the introductory content or a container for a group of introductory content.

- “<nav>”: Represents a section of a page that contains navigation links.

- “<article>”: Represents a self-contained composition, such as a blog post, news article, or forum post.

- “<section>”: Represents a standalone section of content.

- “<aside>”: Represents content that is tangentially related to the main content, such as sidebars or pull quotes.

- “<footer>”: Represents the footer of a document or a section.

By using semantic tags appropriately, you provide meaningful structure to your web pages, which benefits both users and developers. It improves accessibility, search engine optimization, code organization, and overall user experience.

**Q.3) Differentiate between HTML Tags and Elements?**

**Ans.:**

1. HTML Tags:

HTML tags are the building blocks of an HTML document and are used to define the structure and presentation of content on a web page. They are represented by angle brackets (< >) and consist of an opening tag and a closing tag. The opening tag starts with “<” followed by the tag name, and the closing tag starts with “</” followed by the same tag name. For example:

<p>This is an HTML paragraph.</p>

In this example, “<p>” is the opening tag, and “<p>” is the closing tag, surrounding the content "This is an HTML paragraph." The tags define the paragraph element on the web page.

2. HTML Elements:

HTML elements are made up of HTML tags, their content, and any attributes that provide additional information about the element. An HTML element represents a complete unit, starting with an opening tag, followed by the content, and ending with the closing tag. Using the same example:

<p>This is an HTML paragraph.</p>

Here, the entire entity “<p>This is an HTML paragraph.</p>” is the HTML element, consisting of the opening tag “<p>”, the content "This is an HTML paragraph," and the closing tag “</p>”. The element represents a paragraph on the web page.

**Q.6) What are some of the advantages of HTML5 over its previous versions?**

**Ans.:**

HTML5, the fifth major version of the HTML specification, introduced several advancements and improvements over its previous versions. Here are some advantages of HTML5:

1. Enhanced Multimedia Support: HTML5 introduced native support for audio and video elements, eliminating the need for third-party plugins like Adobe Flash. This made it easier to embed multimedia content directly into web pages, providing better compatibility and improved performance.

2. Mobile Device Support: HTML5 was designed with mobile devices in mind. It introduced features like responsive design, which allows web pages to adapt and display appropriately on different screen sizes. Additionally, HTML5 introduced geolocation APIs, local storage, and offline web application support, enhancing the user experience on mobile devices.

3. Semantic Elements: HTML5 introduced several new semantic elements such as `<header>`, `<nav>`, `<section>`, `<article>`, `<footer>`, and more. These elements provide clearer structure and meaning to the content, improving accessibility, search engine optimization, and code readability.

4. Canvas and WebGL: HTML5 introduced the `<canvas>` element, which allows dynamic rendering of graphics, animations, and interactive visualizations directly within the browser without the need for plugins. Additionally, HTML5 introduced WebGL, a JavaScript API for rendering 3D graphics, providing a powerful tool for creating interactive and visually rich experiences on the web.

5. Form Enhancements: HTML5 introduced new form input types and attributes, such as `input type="email"`, `input type="date"`, `input type="range"`, and more. These enhancements made it easier to develop and validate forms, improving the user experience and reducing the reliance on JavaScript for form handling.

6. Improved APIs and Offline Capabilities: HTML5 introduced new JavaScript APIs, including drag and drop, web storage (localStorage and sessionStorage), history manipulation, web workers, and more. These APIs enable developers to create rich, interactive web applications with improved performance and offline capabilities.

7. Backward Compatibility: HTML5 was designed to be backward compatible, meaning older web pages written in previous versions of HTML should still work correctly in HTML5-supporting browsers. This allows developers to gradually transition their existing web pages to HTML5 without breaking compatibility with older browsers.

**Q.8) What is the difference between <figure> tag and <img> tag**

**Ans.:**

The <figure> tag and <img> tag are both used to display images in HTML, but they have different purposes. The <figure> tag is used to represent self-contained content, such as an image with a caption, while the <img> tag is used to simply insert an image.

The <figure> tag can be used to group related content together, such as an image and its caption, or a video and its transcript. This can make it easier for screen readers and other assistive technologies to understand the content of the page.

The <img> tag, on the other hand, does not provide any additional semantics or structure beyond displaying the image. This means that it is not as accessible as the <figure> tag.

Here is an example of how the <figure> and <img> tags can be used:

<figure>

<img src="image.jpg" alt="Description of the image">

<figcaption>A beautiful landscape photo.</figcaption>

</figure>

In this example, the <figure> tag is used to wrap around the <img> tag and the <figcaption> tag. This creates a self-contained unit representing an image with its caption.

The <img> tag is used to insert the image itself. The src attribute specifies the source of the image, and the alt attribute provides alternative text for the image. This text is displayed if the image cannot be displayed for some reason.

**Q.9) What’s the difference between html tag and attribute and give example of some global attributes?**

**Ans.:**

HTML tags are used to define the structure and content of an HTML document. They consist of an opening tag, content, and closing tag. Attributes are placed within the opening tag of an HTML element and provide additional information or modify the behavior of the element.

Some examples of global attributes that can be used with any HTML element include `class`, `id`, `style`, `title`, `lang`, `accesskey`, and `tabindex`. These attributes can be used to specify CSS classes, an identifier, inline CSS styles, a tooltip, the language of the element's content, a keyboard shortcut, and the tab order of an element.

For example, the `<p>` tag can use the `class`, `id`, and `title` attributes to specify CSS classes, an identifier, and a tooltip for the paragraph. The `<a>` tag can include the `href`, `class`, and `data-\*` attributes to define a link's destination, CSS classes, and custom data. The `<img>` tag can utilize the `src`, `alt`, `style`, and `tabindex` attributes to specify the image source, alternative text, inline CSS styles, and tab order.