Module-5 - HTML-5

HTML - 5

THEORY ASSIGNMENT

Question 1 : Difference B/W HTML & HTML5?

HTML (HyperText Markup Language) and HTML5 are both versions of the same language used to structure content on the web, but HTML5 is the latest and most advanced version. Here's a comparison between the two:

1. Doctype Declaration:

- **HTML:** In HTML, the doctype declaration is longer: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
- **HTML5:** In HTML5, the doctype declaration is simpler and more concise: <!DOCTYPE html>

2. Elements and Structure:

- **HTML:** HTML uses older elements like <font>, , <big>, etc., which are now considered outdated for modern web design.
- **HTML5:** HTML5 introduces new semantic elements like <article>, <section>, <nav>, <header>, <footer>, <figure>, <aside>, etc. These elements improve the structure of the web page and enhance accessibility.

3. Multimedia Support:

- **HTML:** HTML requires external plugins like Flash or QuickTime to play multimedia content such as audio or video.
- **HTML5:** HTML5 supports native multimedia elements, such as <audio> and <video>, allowing for easier integration of audio and video content without the need for third-party plugins.

4. Forms:

- **HTML:** HTML supports basic form elements like <input>, <select>, and <textarea>, but lacks some of the advanced input types and attributes.
- **HTML5:** HTML5 introduces new input types such as email, tel, number, date, url, etc., which help with form validation. It also adds new attributes like placeholder, required, autofocus, and pattern.

5. APIs and Storage:

- **HTML:** HTML lacks modern JavaScript APIs for client-side functionality.
- **HTML5:** HTML5 includes powerful JavaScript APIs such as Geolocation, Local Storage, Session Storage, Web Workers, and the Canvas API, enabling more interactive, dynamic, and efficient web applications.

6. Canvas and SVG:

- **HTML:** Older versions of HTML lacked support for drawing graphics directly on the web page.
- **HTML5:** HTML5 introduces the <canvas> element, which allows dynamic, scriptable rendering of 2D shapes, bitmap images, and animations. HTML5 also improves SVG (Scalable Vector Graphics) support.

7. Compatibility and Browser Support:

- **HTML:** Older HTML versions were less compatible with modern web features.
- **HTML5:** HTML5 is more compatible with modern web browsers, enabling new features like offline web apps, geolocation, and local storage.

8. Deprecated and Removed Elements:

- **HTML:** Older versions of HTML used tags like <font>, , <marquee>, etc., which were used for styling purposes.
- **HTML5:** HTML5 removes or deprecates these tags in favor of CSS for styling and more semantic HTML for structure.

9. Mobile-Friendly:

- **HTML:** Older versions of HTML were not designed with mobile responsiveness in mind.
- **HTML5:** HTML5 is more mobile-friendly and optimized for modern, responsive design practices.

10. Character Encoding:

- **HTML:** In HTML, developers had to specify character encoding manually in the head of the document.
- **HTML5:** HTML5 assumes UTF-8 encoding by default, making it easier to work with international characters.

Question 2 : what are the additional tags used in HTML5?

HTML5 introduced several new tags (also called elements) that help improve the structure, accessibility, and functionality of web pages. These tags focus on making web content more semantic, interactive, and responsive. Here's a list of **additional tags** introduced in HTML5:

1. Structural and Semantic Elements:

- **<header>**: Represents the header of a page or section. It can contain navigation links, logos, or introductory content.
- **<footer>**: Defines the footer for a document or section, typically containing copyright information, contact details, or links.
- **<article>**: Specifies independent, self-contained content (e.g., blog posts, news articles).
- **<section>**: Represents a thematic grouping of content, typically with a heading. Used for grouping related content.
- **<nav>**: Defines a section of navigation links (e.g., a menu).
- **<aside>**: Represents content that is tangentially related to the content around it, such as sidebars or advertisements.
- **<main>**: Represents the dominant content of the <body> element, excluding headers, footers, and sidebars. There should only be one <main> element per page.
- **<figure>**: Represents content like images, illustrations, charts, or videos, along with a caption (<figcaption>).
- **<figcaption>**: Defines a caption for the <figure> element.
- **<mark>**: Highlights or marks text, often used for search results or emphasized content.
- **<progress>**: Displays the progress of a task, such as a download or file upload.
- **<meter>**: Represents a scalar measurement within a known range, such as disk usage or a temperature scale.

2. Media Elements:

- **<audio>**: Embeds sound content on a web page, such as music or podcasts. It supports various audio formats and includes controls (play, pause, volume).
- **<video>**: Embeds video content, allowing playback without third-party plugins like Flash. It supports various video formats and includes controls (play, pause, volume, etc.).
- **<source>**: Specifies multiple media resources for <audio> and <video>, allowing the browser to choose the appropriate format.
- **<track>**: Provides text tracks for <video> and <audio> elements, such as subtitles or captions.

3. Form Elements:

- **<input> types** (new types introduced in HTML5):
    - email: For email addresses.
    - tel: For telephone numbers.
    - number: For numeric input.
    - date: For selecting a date.
    - url: For entering a web URL.
    - search: For search input.
    - range: For specifying a range of values.
    - color: For selecting a color (color picker).
    - datetime-local: For selecting both date and time.
- **<datalist>**: Provides a list of predefined options for an <input> element.
- **<output>**: Represents the result of a calculation or user action (e.g., a form submission).

- **<keygen>**: Used to generate a key pair for secure forms (now deprecated but still supported in some browsers).
- **<fieldset>**: Groups related form elements, often used with <legend> to describe a section of the form.
- **<legend>**: Provides a title for the <fieldset>.

4. Scripting and API Elements:

- **<canvas>**: Allows for dynamic rendering of 2D shapes, bitmap images, and animations directly on a web page, often used for graphics and game development.
- **<svg>**: Supports Scalable Vector Graphics for high-quality, resolution-independent graphics like icons and diagrams.
- **<math>**: Enables the embedding of mathematical formulas and expressions.
- **<noscript>**: Specifies content to display if JavaScript is not supported or disabled in the browser.
- **<iframe>** (enhanced features): Used for embedding external content (like videos, maps, etc.) in a webpage, with enhanced security and flexibility in HTML5.

5. Miscellaneous:

- **<wbr>**: Suggests a line break opportunity within a word (word break hint) to improve text wrapping.
- **<template>**: Represents a reusable block of content that is not rendered when the page loads but can be used later via JavaScript.
- **<data>**: Associates a machine-readable value with content that can be displayed as normal text.

- **<bdi>**: Isolates a portion of text that may have different text direction from the surrounding content, ensuring proper text flow (useful for internationalization).
- **<bdo>**: Specifies the direction of text flow, used for changing text direction (left-to-right or right-to-left).

6. New Elements for Accessibility and Performance:

- **<details>**: Defines a disclosure widget from which the user can view or hide additional content.
- **<summary>**: Provides a summary or heading for a <details> element, used in conjunction with it to show or hide content.
- **<dialog>**: Represents a dialog box or popup window, which can be shown or hidden dynamically with JavaScript.