HTML_5

Difference b/w HTML & HTML5?

Feature	HTML	HTML5
Doctype Declaration	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" ...>	<!DOCTYPE html>
New Elements	Limited to basic elements like , , etc.	Introduced new elements like <article>, <section>, <nav>, <header>, <footer>, <aside>, and more for better structure.
Multimedia Support	Requires third-party plugins like Flash for multimedia.	Native support for audio and video with <audio> and <video> tags.
Forms	Forms are limited in functionality.	Enhanced forms with new input types (e.g., email, date, range), and attributes like placeholder, required, etc.
JavaScript APIs	Limited support for JavaScript APIs.	Introduced new APIs for local storage, geolocation, offline web apps, canvas, web workers, etc.
Semantics	Lacks semantic structure for better accessibility.	Improved semantic structure with elements like <figure>, <mark>, <progress>, <output>, and more.
Browser Compatibility	Supported by all browsers but less optimized for modern web apps.	Designed to be supported by modern browsers, though some features may need polyfills for older browsers.
SVG Support	Limited or no native support for SVGs.	Full support for SVG graphics, allowing scalable vector graphics directly in HTML.
Offline Storage	No native support for local storage.	Supports local storage and session storage for saving data on the client side.
Mobile Optimization	Less focus on mobile support.	Optimized for mobile devices, with better responsiveness and support for touch events.
Deprecated Elements	Some elements like <font>, , etc., are allowed.	Many outdated elements (e.g., <font>,) and attributes (e.g., bgcolor) are deprecated or removed.

HTML5 introduces a more semantic, feature-rich, and mobile-friendly version of HTML, with significant improvements in multimedia, forms, storage, and accessibility compared to the older versions of HTML.

What are the additional tags used in HTML5?

HTML5 introduced several new tags to improve the structure, semantics, and functionality of web pages. Here are the key additional tags used in HTML5:

Tag	Description
<article>	Represents a self-contained piece of content (e.g., a blog post or news article).
<section>	Defines a section in a document, typically for grouping related content.

Tag	Description
<nav>	Defines a navigation block for links to other pages or sections.
<header>	Represents the header of a document or section, often containing introductory content or navigational links.
<footer>	Represents the footer of a document or section, typically containing metadata or copyright information.
<aside>	Represents content that is tangentially related to the content around it (e.g., sidebars).
<figure>	Represents self-contained content, such as an image, illustration, or chart, typically with a caption.
<figcaption>	Provides a caption for the content inside a <figure> element.
<mark>	Highlights text that is relevant or has special importance (e.g., search results).
<progress>	Represents the progress of a task or operation, such as a file upload or download.
<output>	Represents the result of a calculation or user action, often used with forms.
<meter>	Represents a scalar measurement within a known range, such as a gauge or progress bar.
<canvas>	Provides a space for rendering graphics on the fly using JavaScript, such as drawings or animations.
<datalist>	Defines a list of predefined options for an <input> element, providing suggestions as the user types.
<keygen>	(Deprecated in HTML5.2) Used for generating key-pairs for forms, typically in cryptography.
<output>	Represents the result of a calculation or user action, such as a form's output value.
<video>	Embeds a video file on the web page, with support for controls like play, pause, and volume.
<audio>	Embeds an audio file, supporting controls like play, pause, and volume.
<source>	Specifies multiple media resources for <video> or <audio> elements, allowing different formats for compatibility.
<track>	Provides text tracks for <video> and <audio>, such as subtitles or captions.

These HTML5 tags help to structure content more meaningfully, provide better accessibility, and support modern multimedia features like video, audio, and dynamic graphics. They also enhance semantic clarity, making the code more readable and SEO-friendly.