DIFFERENCE BETWEEN HTML4 & HTML5

HTML4	HTML5
o It doesn't support audio and video without the use of flash player support	o It support audio and video controls with the use of audio and video tags.
o It is supported by all browsers	o It is supported in some browsers only
o Multimedia supporting tags are not available.	o Multimedia supporting tags are available.
o Difficult for developers to handle inaccurate syntax errors.	o It can handle inaccurate syntax errors.
o To store important data on the client-side, the browser's cache is used.	o HTML 5 uses the web SQL database, web storage, and application cache to store offline data that can be accessed through HTML5 Java script interface
o \<div\> tag is used for separating two parts.	o Does not require \<div\> tag.

Features of HTML5

> **Semantic Elements**

When constructing an HTML script, the term "**semantic**" refers to the meaning of the text. Semantics, to put it simply, is the use of HTML tags that accurately reflect the content they contain. For instance, using the **\<div\>** tag doesn't suggest the content it will carry, however using the **\<p\>** tag suggests it can be used to hold paragraph information.

> **Audio and Video Support**

One of the many HTML 5 features is the support for audio and video. It has reduced the hassle of relying upon third-party services such as Adobe Flash player. To embed Audio and Video into your HTML document, you may use the following two**,\<audio\>** tags and**\<video\>** tags.

These two tags are launched with the release of HTML 5 and support a numerous range of attributes such as height, width, and more that offers developers to leverage the customization of HTML documents.

➢ **Canvas Elements**

Canvas Elements is a top-notch feature that has made the tedious task of handling graphics easier for developers. With the help of Canvas elements, you can easily draw graphics using javascript.It is optimum for creating simple animations and drawing photo compositions.

To initialize the Canvas element **<canvas>** tag is used with only two specific attributes, width and height. Once, the element is initialized, it is then taken care of with the help of JavaScript to draw shapes, lines, images, and more.

➢ **Geolocation API**

The API Geolocation is an HTML feature that is used to access the geographical position of a user, however, it is not accessed unless the user approves of it. If you're wondering where this feature is useful, these come in handy while creating apps like taxi apps, food order tracking apps, fitness tracking apps, and more.

➢ **Local Storage**

It is a modern feature of HTML and several browsers that typically store data in the user's browsers and can access them with the help of JavaScript APIs. This feature is useful for creating offline applications where data is need to be stored locally. Moreover, using this feature you can reduce the transactions between the application and the backend server, creating a fast application. However, there are some limitations, such as there is limited storage, and you can not access more storage than that.

➢ **Responsive Images**

Earlier to create responsive images on the web, you have to rely upon several lines of CSS and sometimes JavaScript, however, HTML 5 makes the process handy by including **srcset** attribute to specify multiple versions of an image at different screen resolutions.

➢ **Web Sockets**

In the previous versions of HTML, when a client sends a request to the backend server, the server then responds afterward. However, in HTML 5 we can establish a bidirectional live communication between the server and the client (a web browser) to reduce the latency in the responses. It has its usage in large-scale applications due to its capability of handling large amounts of connections simultaneously between the server and the client such as online gaming applications, live trading, and other live low-latency applications.

➢ **Cross Document Messaging**

In general, web browsers don't let web pages from different domains influence each other, this is done for several security reasons, however, even if the web pages don't intend to intrigue the privacy or harm the other web page. There are several scenarios, in which you want to access cross-document communication to make your web page more interactive with less effort