# HTML 5

The term HTML5 is essentially a buzzword that refers to a set of modern web technologies. This includes the HTML Living Standard, along with JavaScript APIs to enhance storage, multimedia, and hardware access.

## Advantages of HTML 5:

**1. Cleaner markup/ Improved code:** HTML5 will enable web designers to use cleaner, neater code. We can remove div tags and replace them with semantic HTML5 elements.

**2. Elegant forms:** HTML5 enables designer to use fancier forms. There will be different type of inputs, search and different fields for different purpose.

**3. Consistency:** As websites will adopt the new HTML5 elements we will see more consistency in terms of HTML used to code a web page on one site compared to another. This will make it much easier for designers and developers to immediately understand how a web page is created.

**4. Supports rich media elements:** HTML5 has an inbuilt capability to play audio and video and so we can bid goodbye to those plugin tags.

**5. Offline Application Cache:** The Application Cache concept means that a web application is cached. It can be accessible without the need for internet connection. Some advantages of Application Cache:

1. Offline browsing – Web users can also use the application when they are offline.
2. Speed – Cached resources load quicker
3. Reduce the server load – The web browser will only download updated resources from the server.

**6. No more cookies** – As HTML 5 uses web storage, there is no need of using cookies.

## HTML 5 features:

Html 5 has introduced some new features they are:

1.semantic elements
2. forms
3.web storage
4. canvas,
5. audio and video,
6. Geolocation
7.web socket
8. server sent events (SSE),
9.micro data
10.drag and drop.

**New semantic elements:** A semantic element clearly describes its meaning to both the browser and the developer.
The example of the non-semantic elements are 1. , 2. they tells nothing about its contents.
The examples of semantic elements are 1.<form>,2.<table>,3.<article>they clearly define its content.

**Forms:** uses forms 2.0 to facilitate UI design. Forms are used to collect the data from the user. In HTML file form has have more semantic elements. In form 2.0 in addition to form input types and several new input types are added.

The following new input types are added 1. datetime-local ,2. month, 3. week, 4. date, 5. time, 6. number, 7. range, 8. email, 9.url.

The <output> tag is introduced to display the output. The attributes placeholder, autofocus and required are added.

**Web storage API** (1. local storage 2. session storage)
Web storage API provides mechanisms by which browsers can store key/value pairs in a much better way than using cookies. the two mechanisms within web storage are as follows:

1.**SessionStorage:** maintenance a separate area for each given origin that's available for the duration of the page session (as long as the browser is open including page reloaded and restores). Stores data only for a session meaning that the data is stored until the browser (or tab) is closed. the storage limit is larger than a cookie (at most 5MB).

2.**localStorage:** does the same thing but persists even when the browser is closed and reopened. Stores data with no expiration date and gets clear only through java script or cleaning the browser cache /locally stored data. Storage limit is larger than the cookie and session storage.

Two main benefits of HTML5 Web Storage:

- It can store up to 10 MB data which is certainly more than what cookies have.
- Web storage data cannot be transferred with the HTTP request. It helps to increase the performance of the application.

window.localStorage and window.sessionStorage are the two methods in which methods will be present in HTML5.

- **window.localStorage:** stores data with no expiration date.
- **window.sessionStorage:** stores data for one session

**Canvas:** it allows drawing the user interface. Compare to SVG, it is more effective and user friendly. It allows java script code to perform 2D drawing. <canvas> is the tag to be used in association with java script.

**Audio & video:** HTML 5 supports <audio> and <video> tags for involving multimedia content in the HTML pages.

**Geolocation API:** The HTML Geolocation API is used to get the geographical position of a user. Since this can compromise privacy, the position is not available unless the user approves it.

**Web socket:** Web Sockets is a next-generation bidirectional communication technology for web applications which operates over a single socket and is exposed via a JavaScript interface in HTML 5 compliant browsers.

Once you get a Web Socket connection with the web server, you can send data from browser to server by calling a send() method, and receive data from server to browser by an onmessage() event handler.

**Server sent events (SSE):** A server-sent event is when a web page automatically gets updates from a server. This was also possible before, but the web page would have to ask if any updates were available. With server-sent events, the updates come automatically.

Examples: Facebook/Twitter updates, stock price updates, news feeds, sport results, etc.

**Micro data:** Microdata is a standardized way to provide additional semantics in your web pages. Microdata lets you define your own customized elements and start embedding custom properties in your web pages.

At a high level, microdata consists of a group of name-value pairs. The groups are called items, and each name-value pair is a property. Items and properties are represented by regular elements.

**Drag and drop API:** In HTML, any element can be dragged and dropped. Drag and drop is a very common feature. It is when you "grab" an object and drag it to a different location. This feature is basically related to data transfer feature in HTML5. It is implemented by associating HTML with java script.

In fact, some of the above feature's HTML are implemented through API. They are:

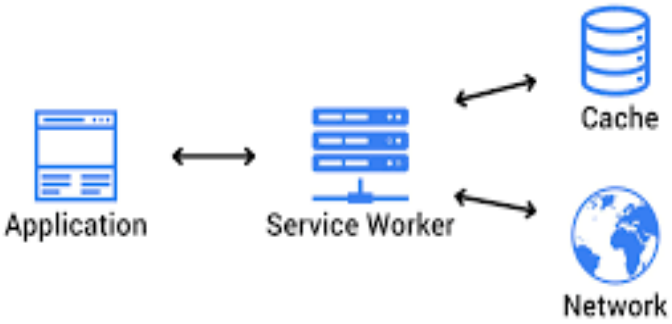
- Geolocation API
- Drag/Drop API
- Web storage API
- Web workers API
- Web SSE

**Web workers:** Web workers make it possible to run a script operation in a background third separate from the main execution thread of a web application. Advantage of this is that laborious processing can be performed in a separate thread along the main (usually the UI) thread to run without being blocked/slowed down.

There are two different types of web workers in HTML5 i.e., Dedicated Workers and Shared Workers.

**Service workers:**

Service workers essentially act as proxy servers that sit between web applications, the browser, and the network (when available). They are intended, among other things, to enable the creation of effective offline experiences, intercept network requests and take appropriate action based on whether the network is available, and update assets residing on the server. They will also allow access to push notifications and background sync APIs.




**Some of the tags that are removed from the HTML 5 are:**

<applet>, <font>, <frame>, <frameset>, <noframe>, <big>, <acronym>, , <tt>.

**HTML5 tags are classified into Semantic and Non-semantic elements.**

1. Non-semantic elements are: ,  and <p>.
2. Semantic elements have meaningful names which tells about type of content. For example, header, footer, table, ... etc. HTML5 introduces many semantic elements as mentioned below which make the code easier to write and understand for the developer as well as instructs the browser on how to treat them.

    - article
    - aside
    - details
    - figcaption
    - figure
    - footer
    - header
    - main
    - mark
    - nav
    - section

**header:** The **<header>** HTML element represents introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also a logo, a search form, an author name, and other elements.

```
<header>
  <h1>Main Page Title</h1>
  <img src= “college.png" alt="My College logo">
</header>
```

**Footer:** The **<footer>** HTML element represents a footer for its nearest sectioning content or sectioning root element. A <footer> typically contains information about the author of the section, copyright data or links to related documents.

```
<article>
   <h1>How to be a Professional</h1>
   <ol>
      <li>Be strong in Academics</li>
      <li>Be strong in Problem Solving</li>
      <li>Be an active member in SAC, IUCEE, EWB, IEC, UIF</li>
   </ol>
   <footer>
      <p>© 2022 VVIT, Nambur</p>
   </footer>
</article>
```

**article:** The **<article>** HTML element represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication). Examples include: a forum post, a magazine or newspaper article, or a blog entry, a product card, a user-submitted comment, an interactive widget or gadget, or any other independent item of content.
Ex:

```
<article>

  <header>
   <h2>The Planet Earth</h2>
   <p>
         Posted on Wednesday,
         <time datetime="2022-03-22">22nd March 2022</time>
         by CSE Dept</p>
  </header>

  <p>We live on a planet with so many things still unseen</p>
```

```
  <p><a href="https://example.com/the-planet-earth/">
          Continue reading....
  </a></p>

</article>
```

**Aside:** The **<aside>** HTML element represents a portion of a document whose content is only indirectly related to the document's main content. Asides are frequently presented as sidebars or call-out boxes.

Ex:

```
<p>Vasireddy Venkatadri Institute of Technology (VVIT) was established in the year 2007, with an intake of 240 students in four B. Tech programs under Social Educational Trust in Nambur village, Guntur, AP, by Er. Vasireddy Vidya Sagar
</p>

<aside>

   <p>VVIT is one among the top colleges in A.P region in securing more placements</p>

</aside>

<p>Realizing the slogan that “Everyone Counts,” VVIT has the unique achievement of having placed most its students in MNCs like Tech Mahindra, Infosys, TCS, Wipro, etc. (almost 50 companies) every year</p>
```

**nav:** The **<nav>** HTML element represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes.

```
<nav class="menu">
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>
```

**CSS File:**

```
aside {
   width: 40%;
   padding-left: .5rem;
   margin-left: .5rem;
   float: right;
```

```
    box-shadow: inset 5px 0 5px -5px #29627e;
    font-style: italic;
    color: #29627e;
}
aside > p {
    margin: .5rem;
}
p {
    font-family: 'Fira Sans', sans-serif;
}
```

**Section:** The **<section>** HTML element represents a generic standalone section of a document, which doesn't have a more specific semantic element to represent it. Sections should always have a heading, with very few exceptions.

Ex:

```
<h1>Choosing a right JOB </h1>
<section>
    <h2>Introduction</h2>
    <p>This document provides a guide to help with the important task of choosing the correct Job</p>
</section>
<section>
    <h2>Criteria</h2>
    <p>There are many different criteria to be considered when choosing a JOB — Salary, working hours, place, etc...</p>
</section>
```

**Video & Audio:** Before HTML 5 came into existence, videos could only be played in a browser using a plugin like flash. But after the release of HTML 5, adding a video and an audio to a webpage is as easy as adding an image. There are three different formats that are commonly supported by web browsers – mp4, Ogg, and WebM.

Syntax: <video src="" controls>  </video>

Video Ex:

```
<!DOCTYPE html>
<html>
<body>
 <p>Adding Video on my webpage </p>
    <video width="400" height="350" controls>
```

```
    <source src="myvid.mp4" type="video/mp4">
    <source src="myvid.ogg" type="video/ogg">
  </video>
</body>
</html>
```

**Attributes that can be used with the "video" tag are listed below:**

1. **Autoplay:** It tells the browser to immediately start downloading the video and play it as soon as it can.
2. **Preload:** It intends to provide a hint to the browser about what the author thinks will lead to the best user experience.
3. **Loop:** It tells the browser to automatically loop the video.
4. **height:** It sets the height of the video in CSS pixels.
5. **width:** It sets the width of the video in CSS pixels.
6. **Controls:** It shows the default video controls like play, pause, volume, etc.
7. **Muted:** It mutes the audio from the video.
8. **Poster:** It loads an image to preview before the loading of the video.
9. **src:** It is used to specify the URL of the video file.

**Audio:** The HTML <audio> element is used to play an audio file on a web page.

**Syntax/ex:**

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
   Your browser does not support the audio element.
</audio>
```

**Canvas:** The HTML <canvas> element is used to draw graphics on a web page.

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript. The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Ex-1: <canvas id="myCanvas" width="200" height="100"></canvas>

Ex-2: <canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;"> </canvas>

**SVG:** Earlier <svg> is used. The HTML <svg> element is a container for SVG graphics. SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

Ex:

```
<!DOCTYPE html>
<html>
<body>
<svg width="100" height="100">
<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>
</body>
</html>
```

**SVG Vs canvas:**

| Canvas | SVG |
|---|---|
| 1.pixel graphics. | 1.vector graphics. |
| 2.Resolution dependent. | 2.Resolution independent. |
| 3.drawn with java script. | 3.XML format. |
| 4.modified through script only. | 4.modified through css and script. |
| 5.not suitable for printing at high resolution. | 5.high quality print at any resolution. |
| 6.suitable for games. | 6.not suitable for games. |

******