HTML

HTML (Hyper Text Markup Language) is a markup language for creating web pages. Presently, the latest version of HTML is called HTML5.

HTML element generally starts with an opening tag and ends with a closing tag.

<h1> Hello</h1>

Almost every element allows attributes like **id, style, class, title**. Attribute defines an element.

<a href=”http://google.com” id=”link1”>Google</a>

Structure of the HTML page:

```
<!DOCTYPE html>
<html>
      <head>
      </head>
      <body>
      </body>
</html>
```

<!DOCTYPE html> is used for declaring HTML5 page (can also be used for declaring HTML4 page).

<head> is optional in HTML page.

Block elements like **<div>** tag takes up the area from left to right of page.

Hyperlinks are created using anchor **<a>** tag. It has a **target** attribute that tells the link to open in following different ways:

- _blank : new window or tab
- _parent : in the parent frame
- _self : current frame (default)
- _top : in the body of its window
- framename : in the specified frame

Links can be linked to another links in the same page as below:

<a id=”Top”>Top</a>

<a href=”#Top”>Go to top</a>

Images are created using **<img />** tag. **src** attribute of **img** tag is used to link any uploaded image. **alt** attribute of **img** tag shows an alternate text if the image does not load up.

Sub script (as in H_2O) can be created as shown below:

H<sub>2</sub>O

Computer code can be placed inside **<code>** tag, so that it will appear in a different font.

Preformatted element <pre> is used to preserve line breaks and spaces in a text.

Almost every single HTML element uses a style attribute for styling and it takes in CSS as input.

HTML comments are created as:

<!—This is an HTML comment -->

To put two div elements beside each other (same line), we use:

<div style=”display:inline-block;”>Block 1</div>

<div style=”display:inline-block;”>Block 2</div>

Lists can be created as:

```
<ul>
      <li>First item</li>
      <li>Second item</li>
      <li>Third item</li>
</ul>
```

<ul> stands for unordered list. Following list gets created.

- First item
- Second item
- Third item

Similarly, ordered list can be created using **<ol>** tag.

Definition Lists can be created as follows:

```
<dl>
      <dt>Definition Title</dt>
      <dd>Definition Description</dd>
</dl>
```

It comes up as follows:

Definition Title

      Definition Description

Tables are created as shown below:

```
<table>
<tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Major</th>
</tr>
<tr>
    <td>Apurva</td>
    <td>Sinha</td>
    <td>IT</td>
</tr>
</table>
```

It comes up as follows:

First Name	Last Name	Major
Apurva	Sinha	IT

Sample usage of HTML form:

```
<!DOCTYPE html>
<html>
<body>
<form method="get" action="http://google.com">
<table columns=2 cellpadding=5>
<tr>
    <td colspan="2" style="text-align: center;"><h3>Sample Form</h3></td>
<tr>
<tr>
```

```
    <td>Name:</td>
    <td><input type="text" name="myName" /></td>
</tr>
<tr>
    <td>Password:</td>
    <td><input type="password" name="userPass" /></td>
</tr>
<tr>
    <td>Gender:</td>
    <td>
        <select name="gender">
            <option value="NA" disabled="disabled"
selected="selected">Select Gender</option>
            <option value="male">Male</option>
            <option value="female">Female</option>
        </select>
    </td>
</tr>
<tr>
    <td>Biography:</td>
    <td><textarea name="bio" rows="5" cols="60">Add a
biography</textarea></td>
</tr>
<tr>
    <td>Major:</td>
    <td>
        <input type="radio" name="major" id="rd1" value="IT" >IT</input>
        <input type="radio" name="major" id="rd2" value="CS" >CS</input>
        <input type="radio" name="major" id="rd3"
value="Maths">Maths</input>
    </td>
</tr>
<tr>
    <td>Courses:</td>
    <td>
        <input type="checkbox" name="checkbox" id="checkbox1" value="DBMS"
>Database Systems</input>
        <input type="checkbox" name="checkbox" id="checkbox2"
value="WEBDEV" >Web Application Development</input>
    </td>
</tr>
<tr>
    <td colspan="2" style="text-align: center;">
        <button type="reset" style="margin-left: 10px;">Reset</button>
        <button type="submit" style="margin-left: 10px;">Submit</button>
    </td>
</tr>
</table>
</form>
</body>
</html>
```

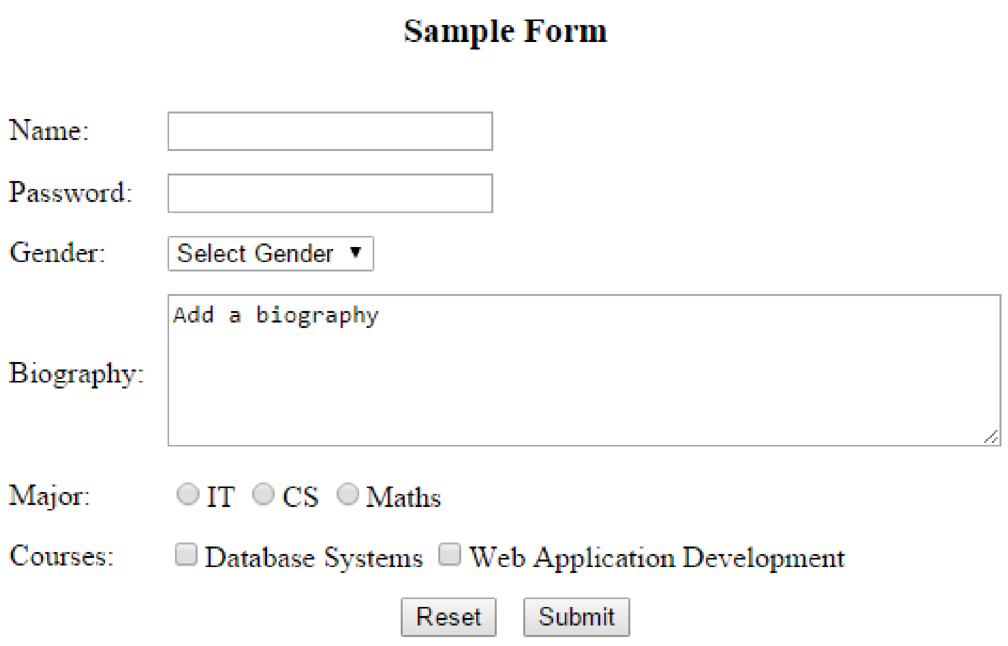


Fig 1.1: HTML Sample Form

Head section of HTML contains the following elements:

- Title – name present in the tab of the browser

  <title>My Page</title>

- Link – links to external CSS files and page icons

  <link rel=”stylesheet” type=”text/css” href=”style.css” />

  <link rel=”icon” type=”image/x-icon” href=”favicon.ico” />

- Script – to link an external JavaScript file

  <script src=”js/myJavaScript.js” />

- Style – to add CSS to the web page

Head section also contains ‘meta’ tags that provide useful information to the search engines regarding the content of the website. Two important types of ‘meta’ tags are ‘keyword’ and ‘description’.

<meta name=”description” content=”This page contains our products and services” />

<meta name=”keyword” content=”database, middleware, consulting” />

HTML 5

New HTML5 elements like **<header>** and **<footer>** (similar to <div> element) helps form a better “Document Object Model” and this would enable search engines to become more accurate in the future. For example, contact information can be put inside the **<address>** element, and search element would be able to find addresses more easily and accurately.

One page can have multiple headers and footer but nesting of such elements (header inside another header) are not allowed.

“spellcheck” and “contenteditable” are two useful features of HTML5. These are global attributes and can be applied to almost any tag. Spellcheck can be used be used along with contenteditable attribute to get spelling suggestions with editable text.

<p spellcheck="true" contenteditable="true">splel chkc this senrance</p>

Spelling corrections are not saved.

HTML5 **<video>** element has eliminated the need of having different plug-in for playing any video content.

```
<video autoplay=”autoplay” controls=”controls” loop=”loop”>
      <source src=”videos.mp4” type=”video/mp4” />
      <source src=”videos.ogg” type=”video/ogg” />
      Your browser does not support this video type
</video>
```

If the browser is unable to read the first video, then it tries to read the next video in the sequence. If it is able to read the first video, then it skips the remaining. If the browser does not support the video tag, then the text given at the end is displayed.

HTML5 <video> element along with JavaScript can provide various video features to the user as shown in the code below:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="margin-left: 100px;">HTML5 Video Element</h1>
```

```
<video id="demo" controls
  src="http://media.w3.org/2010/05/bunny/movie.ogv"
  autoplay=autoplay controls=controls loop=loop
  width="640" height="360" onclick="playOrPause()" >
Your user agent does not support the HTML5 Video element.
</video>
<script type="text/javascript">
    function playOrPause()  {
        if(demo.paused)
            demo.play();
        else
            demo.pause();
    }
</script>
</body>
</html>
```

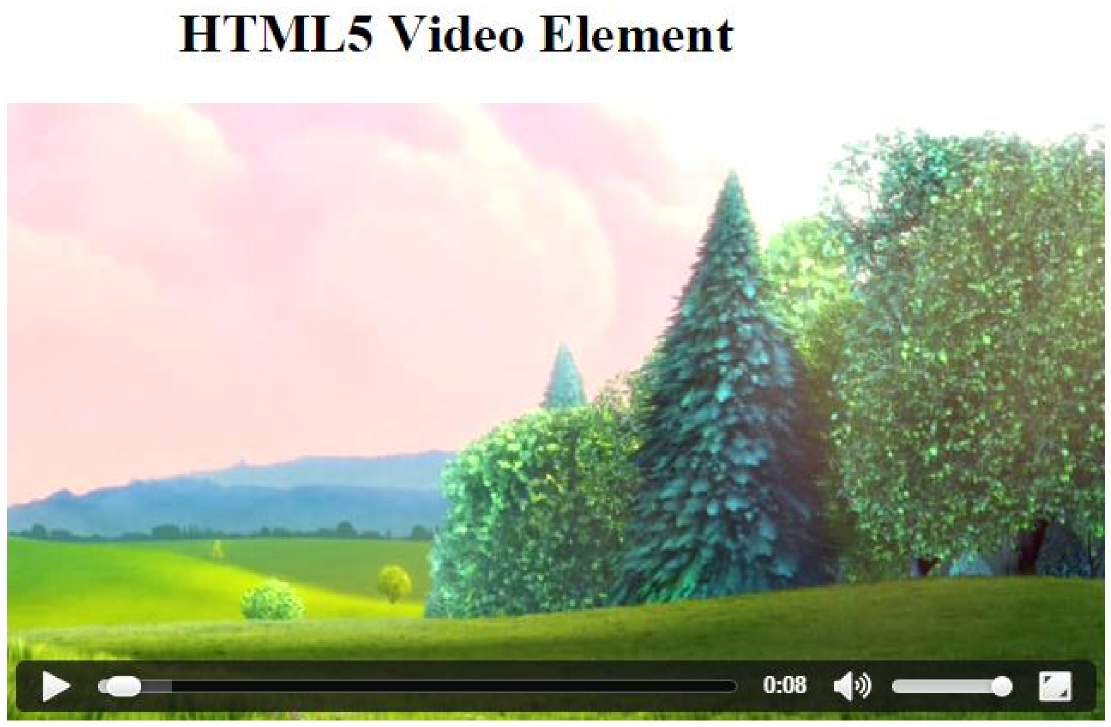

HTML5 Video Element

Fig 1.2: HTML5 Video Element

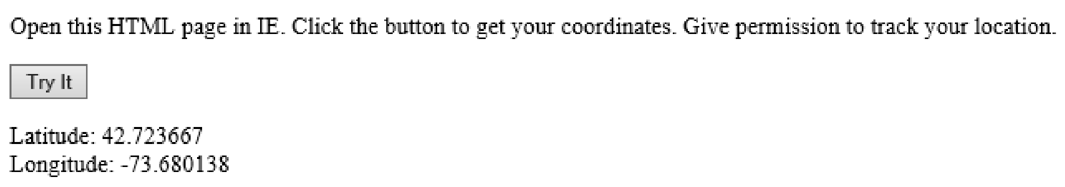
The JavaScript function “playOrPause()” enables user to play or pause the video by clicking anywhere on the video.

<audio> element is very much similar to the **<video>** element and can be used for playing audio without the need of any plug-ins.

HTML5 supports drag and drop feature that allows the user to click an object and drag it into a droppable zone. Event attributes like draggable, ondragstart, ondragover, and ondrop are used to implement drag and drop.

HTML5 supports geo-location. It can return the longitude and latitude of the current position as shown below.

```
<!DOCTYPE html>
<html>
<body>
<p>Open this HTML page in IE. Click the button to get your coordinates.
Give permission to track your location.</p>
<button onclick="getLocation()">Try It</button>
<p id="demo"></p>
<script>
var x = document.getElementById("demo");
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    } else {
        x.innerHTML = "Geolocation is not supported by this browser.";
    }
}
function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
}
</script>
</body>
</html>
```



Open this HTML page in IE. Click the button to get your coordinates. Give permission to track your location.

```
Latitude: 42.723667
Longitude: -73.680138
```

Fig 1.3: HTML5 Geo-location

HTML5 enables user's browser to store its own information with the help of **localStorage** and **sessionStorage**. Difference between localStorage and sessionStorage are:

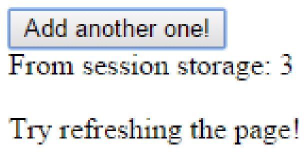
localStorage = data stays until a script from that same website destroys it or the user clears the cache sessionStorage = data stays alive until the window is closed (more temporary)

In the below code, a counter is created using "sessionStorage".

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
if(typeof(Storage) !=="undefined"){
    //we can accept local storage

    //Just load the below line once in the browser, and it will get saved
in the browser's memory
    //Name will be displayed even after commenting out the below line after
page loads for the first time
    localStorage.myName = "Apurva Kumar Sinha";

    alert("From local storage: "+localStorage.myName);
} else {
    //we cannot accept local storage
    alert('No local storage accepted');
}
function counter () {
    if(typeof(Storage) !=="undefined"){
        if(sessionStorage.counter){
            sessionStorage.counter = Number(sessionStorage.counter) + 1;
        } else {
            sessionStorage.counter = 1;
        }
        document.getElementById("number").innerHTML = "From session
storage: "+sessionStorage.counter;
        if(sessionStorage.counter >= 1){
            document.getElementById("number").innerHTML += "<p>Try
refreshing the page!</p>";
        } else {
            document.getElementById("number").innerHTML = "Sorry, your
browser does not support web storage";
        }
    }
}
</script>
</head>
<body>
    <button onclick="counter()" type="button">Add another one!</button>
    <div id="number"></div>
</body>
</html>
```



From session storage: 3

Try refreshing the page!

Fig 1.4: HTML5 Local and Session Storage

Reference: https://www.udemy.com/thecompletewebdeveloper/learn/#/