HTML 5

Created By :- The EasyLearn Academy

www.theeasylearnacademy.com

theeasylearn@gmail.com

Contact No :- 9662512857

What is HTML5?

- HTML5 is the current standard for HTML.
- The previous version of HTML 4.01, came in 1999.
- The web has changed a lot since after that.
- HTML5 introduces several new features and improvements over its predecessors, making it more powerful, flexible, and efficient.
- HTML5 is supported by all modern browsers.
- HTML5 is more device friendly.
- The code becomes cleaner mainly due to replacing div tags with Semantic elements, which help better structure content of the web page and improve readability.
- HTML5 supports geolocation, which makes it possible to determine the users' location.
- The new standards were specified for playing multimedia (animation, audio, video) directly in the browser without having to install additional plug-ins.

New Features

- **New Semantic Elements** − These are like <header>, <footer>, and <section>.
- **Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- **Persistent Local Storage** can store data in local storage without library.
- **Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** − You can embed audio or video on your webpages without resorting to third-party plugins.
- **Geolocation** − Now visitors can choose to share their physical location with your web application.
- **Microdata** − This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- **Drag and drop** − Drag and drop the items from one location to another location on the same webpage.

Html 5 support many API like..

- API for
  - Audio
  - Video
  - Offline application
  - Protocols
  - Editable
  - Drag & drop
- There are some associated API like
  - Geolocation
  - Messaging

Deprecated tag in HTML 5

- Deprecated means you should not use belog tags
  1. Big
  2. Center
  3. tt
  4. **Font**
  5. **u**
  6. **Strike**
  7. basefont
  8. frame
  9. frameset
  10. noframes
  11. applet

The HTML5 <!DOCTYPE>

- Don’t forget to add below line in html 5 standard document.

  <!DOCTYPE html>
- It must be first line in file.
- The doctype tells the browser how to render(process) the page.

Minimum HTML5 Document

- Below is a simple HTML5 document, with the minimum required tags:

```
<!DOCTYPE html>
<head>
<title>Title of the document</title>
</head>

<body>
</body>

</html>
```

HTML <article> Tag

- The article element represents a section of content that forms an independent part of a document or site; for example, a magazine or newspaper article, or a blog entry.
- it is independently distributable or reusable, e.g. in syndication.
- <article> element typically has its own content & header tag and footer tag.

<header> Tag

- The <header> tag specifies a header for a article or for whole web page.
- The <header> element should be used as a container for introductory content or set of navigational links.
- A <header> tag should not be placed within a <footer>, <address> or another <header> element.

<footer> Tag

- The <footer> tag defines a footer for a web page or article.
- A footer normally contains the author of the document, copyright information, links to terms of use, contact information, etc.
- Contact information inside a <footer> element should be in an <address> tag.

example

- Comprehensive example of article tag with its related artical

```
<article>
    <header>
      <h1>HTML5</h1>
      <p>this is content inside the header</p>
    </header>
    <p>this is content inside the artical.</p>
    <footer>
      <p>this is content inside fotter</p>
    </footer>
</article>
```

HTML <aside> Tag

- The HTML <aside> tag is used to represent content that is related to the surrounding content within an article or web page
- This type of content is often represented in sidebars.
- An example is a "pull quote" from a longer article.
- A pull quote is a quotation or edited quotation from an article that is placed in a larger font size on the same page.
- Aside tag contain can highlight a key topic of article.

Example of aside tag

```
<!DOCTYPE html>
<html>
<body>
   <h1>this is heading </h1>
   <artical>
   <aside>
   <h4> Key Points</h4>
   <ol>
     <li> first key points </li>
     <li> secondkey points </li>
   </ol>
   </aside>
   </artical>
</body>
</html>
```

Section tag

- The HTML <section> tag is a semantic HTML tag used to define a section in the document that generally includes a group of related article tag.
- Let us see example

```
<!DOCTYPE html>
<html>
<head>
    <title>Section Example</title>
</head>
<body>
    <section>
        <header>
            <h1>Section Heading</h1>
        </header>
        <article>
            <header>
                <h2>Article 1 Heading</h2>
            </header>
            <p>Content of Article 1 goes here...</p>
            <footer>
                <p>Article 1 Footer</p>
            </footer>
        </article>
        <article>
            <header>
                <h2>Article 2 Heading</h2>
            </header>
            <p>Content of Article 2 goes here...</p>
            <footer>
                <p>Article 2 Footer</p>
            </footer>
        </article>
        <footer>
            <p>Section Footer</p>
        </footer>
    </section>
</body>
</html>
```

HTML <audio> Tag

- audio is used to embed music into html page.
- Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg:
- Supportable file list.

Browser	MP3	Wav	Ogg
Internet Explorer 9+	YES	NO	NO
Chrome 6+	YES	YES	YES
Firefox 3.6+	YES	YES	YES
Safari 5+	YES	YES	NO
Opera 10+	YES	YES	YES

HTML <audio> Tag

- **MIME Types for Audio Formats**
  - **Format** **MIME-type**
  - MP3 audio/mpeg
  - Ogg audio/ogg
  - Wav audio/wav
- **Example**

```
<audio controls autoplay>
    <source src="horse.mp3" type="audio/mpeg">
    Your browser does not support the audio tag.
  This is called fallback content which appear only
  if browser do not support audio tag at all.
</audio>
```

- To convert file from one form to another form you can use
- https://audio.online-convert.com/

HTML <audio> Tag attributes

Attribute	Value	Description
autoplay	autoplay	Specifies that the audio will start playing as soon as it is ready
controls	controls	Specifies that audio controls should be displayed (such as a play/pause button etc).
loop	loop	Specifies that the audio will start over again, every time it is finished
muted	muted	Specifies that the audio output should be muted
preload	auto metadata none	Specifies if and how the author thinks the audio should be loaded when the page loads
src	*URL*	Specifies the URL of the audio file

How to play audio in case browser do not support audio tag?

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<audio controls >
    <source src="a1.mp3" type="audio/mpeg" />
    <embed src="a1.mp3" type="application/x-mplayer2"
  autostart="true">
</audio>
</body>
</html>
```

<datalist> Tag

- The <datalist> tag is used to give the list of pre-defined options for an <input> element.
- When user start typing, The <datalist> tag is automatically provide an "autocomplete" feature on <input> elements.
- Value of the <input> element's list attribute must match with id attribute of with a <datalist> element.

Example

```
<input type='text' list="course"
  name="txtcourse">
<datalist id="course">
    <option value="BCA">
    <option value="MCA">
    <option value="BBA">
    <option value="B.Com">
    <option value="Other">
</datalist>
```

<details> Tag

- The <details> tag display additional details that the user can view or hide.
- Any sort of content can be put inside the <details> tag.
- By default, the content of the tag is not displayed. In order to display the contents, you must apply the open attribute.
- This tag is commonly presented with a small triangle that twists (or rotates) to show open/closed state.

example

```
<details>
    <summary>Show/Hide me</summary>
    <p>this is information which will
  be shown and hide as per
  requirement</p>
</details>
```

open Specifies that the details should be visible (open) to the user

<figure> Tag

- It is used to markup photo in html document.
- The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
- It is mostly used with figcaption tag.
- Figcaption used to add a caption or explanations to the contents of the <figure> tag.
- The <figcaption> is included in the <figure> and is placed as the first or last child element

```
<figure>
  <img src="https://picsum.photos/400/300"
  alt="Random Image from picsum.photos">
  <figcaption>
          This is a random image from
  picsum.photos
  </figcaption>
</figure>
```

<mark> Tag

- The <mark> tag defines marked text.
- It is used to highlight parts of your text by default using yellow color.
- Example

```
<p>
  Do not forget to watch  <mark>movie</mark> today
  at 8 o clock
</p>
```

<nav> Tag

- The <nav> tag is an HTML5 element that is used to define a navigation section on a webpage.
- It is intended to mark a section of the page that contains navigation links to other pages or to different sections of the same page.
- The <nav> element helps search engines and web browsers identify the primary navigation areas of your website.

Example

```
<nav>
  <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#about">About</a></li>
    <li><a
     href="#services">Services</a></li>
     <li><a
     href="#contact">Contact</a></li>
  </ul>
</nav>
```

<time> Tag

- The <time> tag is an HTML5 element used to represent a specific time or a range of time on a webpage.
- It is useful when you want to add machine-readable date and time information to your content, which can be beneficial for search engines, screen readers, and other user agents.
- The <time> tag can be used in two main ways:
  - **Specific Date and Time:** You can use the <time> tag to represent a specific point in time, such as dates, times, or both. You can include the time in either 24-hour format or 12-hour format with AM and PM indicators.
  - **time range:** The <time> tag can also be used to represent a range of time, such as a start date/time and an end date/time. This is often used to indicate the duration of events or appointments.

example

```
<h1>Upcoming Concert</h1>
  <p>The concert starts at <time datetime="2023-07-31T19:00">7:00 PM on July 31, 2023</time>.</p>

<h1>Store Opening Hours</h1>
  <p>The store is open from <time datetime="2023-07-27T09:00">9:00 AM</time> to <time datetime="2023-07-27T18:00">6:00 PM</time> every weekday.</p>
```

<video> Tag

- The <video> tag specifies video, such as a movie clip or other video streams.
- Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg:
- Internet Explorer 8 and earlier versions, do not support the <video> tag.

Supported fortmat in various browser

Browser	MP4	WebM	Ogg
Internet Explorer 9+	YES	NO	NO
Chrome 6+	YES	YES	YES
Firefox 3.6+	YES	YES	YES
Safari 5+	YES	NO	NO
Opera 10.6+	YES	YES	YES

example

```
<video width="320" height="240" controls>
<source src="movie.mp4" type="video/mp4">
<embed src="movie.mp4" type="application/x-mplayer2" />
</video>
```

- For web browsers, you are basically going to need MP4, WebM and Ogg formats
- Ogg format is specified as type=video/ogg but the video extension is .ogv
- while mobile browsers will use MP4 and 3GP formats.

Optional Attributes

Attribute	Value	Description
autoplay	autoplay	Specifies that the video will start playing as soon as it is ready
controls	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
Height	*pixels*	Sets the height of the video player
loop	loop	Specifies that the video will start over again, every time it is finished
muted	muted	Specifies that the audio output of the video should be muted
Poster	*URL*	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
Src	*URL*	Specifies the URL of the video file
width	*pixels*	Sets the width of the video player

Forms in html 5

- Support for web form 2.0
- Like you can now use date picker, color picker, numeric spinner etc.
- Input field types now include email, url, search etc.
- Put and delete from methods are supported

HTML5 New Input Types

- HTML5 has several new input types for forms. These new features allow better input control and validation.
  - color
  - **date**
  - datetime
  - datetime-local
  - **email**
  - **month**
  - **number**
  - range
  - search
  - **tel**
  - **time**
  - **url**
  - week
- Not all major browsers support all the new input types.
- However, you can already start using them; If they are not supported, they will behave as regular text fields.

```
<form action="/submit_form" method="post">
        <label for="full_name">Full Name:</label>
        <input type="text" id="full_name" name="full_name" placeholder="Enter your full name"
required><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" placeholder="example@example.com"
required><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br>

        <label for="phone">Phone:</label>
        <input type="tel" id="phone" name="phone" placeholder="Enter your phone number"
required><br>

        <label for="birthdate">Birthdate:</label>
        <input type="datetime-local" id="birthdate" name="birthdate" required><br>

        <label for="favorite_color">Favorite Color:</label>
        <input type="color" id="favorite_color" name="favorite_color" value="#ff0000"
required><br>

        <label for="quantity">Quantity:</label>
        <input type="number" id="quantity" name="quantity" min="1" max="100" required><br>

        <label for="file_upload">Upload File:</label>
        <input type="file" id="file_upload" name="file_upload" accept=".jpg, .png, .pdf"
required><br>

        <input type="submit" value="Submit">
        <input type="reset" value="Reset">
    </form>
```

HTML5 New Form Attributes

- HTML5 has several new attributes for <form> and <input>.
- New attributes for <form>:
  - autocomplete
  - novalidate

autocomplete

- The autocomplete attribute specifies whether a form or input field should have autocomplete on or off.
- When autocomplete is on, the browser automatically complete values based on values that the user has entered before.
  - **Tip:** It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa.
  - **Note:** The autocomplete attribute works with <form> and with the types like text, search, url, tel, email,, datepickers, range, and color.

Example

```
<form action="form.php" autocomplete="on">
    First name:<input type="text"
  name="fname"><br>
    Last name: <input type="text"
  name="lname"><br>
    E-mail: <input type="email"
  name="email" autocomplete="off">
<br />
    <input type="submit">
</form>
```

novalidate

- The novalidate attribute is a boolean attribute.
- it specifies that the form-data (input) should not be validated when submitted.
- It means that browser ignore required attribute and do not care whether value is according to type or not
- Example

```
<form action="demo_form.asp" novalidate>
    E-mail: <input type="email"
  name="user_email">
    <input type="submit">
</form>
```

Now let us see attributes of **input tag** in html 5

autofocus

- The autofocus attribute is a boolean attribute.
- When present, it specifies that an <input> element should automatically get focus when the page loads.
- First name:<input type="text" name="fname" autofocus>

<input> formaction Attribute

- The formaction attribute specifies the URL of a file that will process the input control when the form is submitted.
- The formaction attribute overrides the action attribute of the <form> element.
- **Note:**
  - The formaction attribute is used with type="submit" and type="image".
  - Example

```
<form action="teacher_submit.html">
    Email Address: <input type="email"
  name="email"><br>
    Password: <input type="password"
  name="password"><br>
    <input type="submit" value="Teacher Login"><br>
    <input type="submit" formaction="student
  _submit.html"
    value="Student Login">
</form>
```

<input> min and max Attributes

- The min and max attributes specify the minimum and maximum value for an <input> element.
- The min and max attributes works with number, range, date, datetime, datetime-local, month, time and week.
- Example
- Enter a date before 1980-01-01:
  <input type="date" name="bday" min="1979-12-31">

  Enter a date after 2000-01-01:
  <input type="date" name="bday" max="2000-01-02">

  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">

<input> multiple Attribute

- The multiple attribute is a boolean attribute.
- it specifies that the user is allowed to enter more than one value in the <input> element.
- The multiple attribute works with email, and file input types.
- Example
- Select images: <input type="file" name="img" multiple>

<input> pattern Attribute

- The pattern attribute specifies a regular expression that the <input> element's value is checked against.
- The pattern attribute works with input types like **text, search, url, tel, email, and password**.
- Example
  - Country code:
  - <input type="text" name="country_code" pattern="[A-Za-z]{3}" placeholder="Three letter country code">
- To get more pattern use below website
  - http://html5pattern.com/

<input> accept attribute …

- The accept attribute specifies the types of files that the server accepts (that can be submitted through a file upload).
- The accept attribute can only be used with <input type="file">.
- To specify more than one value, separate the values with a comma (e.g. <input accept="audio/*,video/*,image/*" />

```
<form action="output.php">
    <input type="file" name="myphoto" accept="image/*" />
    <input type="submit">
</form>
```

For complete list of MIME type use below url

https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types

<input> placeholder Attribute

- a short description of the expected format).
- The short hint is displayed in the input field before the user enters a value.
- The placeholder attribute works with text, search, url, tel, email, and password.
- Example
  - <input type="text" name="fname" placeholder="First name">

<input> required Attribute

- The required attribute is a boolean attribute.
- When present, it specifies that an input field must be filled out before submitting the form.
- The required attribute works with text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.
- Example
  - Username: <input type="text" name="usrname" required>

HTML5 Geolocation

- HTML5 Geolocation is used to locate a user's position
- The HTML5 Geolocation API is used to get the geographical position of a user.
- To do that one need to use getCurrentPosition() method to get the user's position.
- The getCurrentPosition() method returns an object if it is successful.
- The latitude, longitude and accuracy properties are always returned.

```
<p id="demo">Click the button to get your
  coordinates:</p>
<input onclick="getLocation()" type="button"
  value="get location" />
<script>
 var mylocation = document.getElementById(“demo");
        function getLocation(){
                navigator.geolocation.getCurrentPositi
  on(showPosition);
        }
        function showPosition(location)
        {
            mylocation.innerHTML = "Latitude = " +
  location.coords.latitude + "<br> Longitude = " +
  location.coords.longitude;
        }
</script>
```

HTML 5 Local Storage

- HTML5 introduced a new web storage feature called "localStorage," which allows web applications to store key-value pairs locally in a web browser. The data stored using localStorage remains available even after the browser is closed or the page is refreshed. It provides a simple way to save and retrieve data on the client-side without the need for server-side databases.
- Here's a brief overview of how to use localStorage in HTML5:
- Storing Data: To store data in localStorage, you can use the localStorage.setItem() method. It takes two arguments: the key (a string) and the value (also a string).
- Retrieving Data: To retrieve data from localStorage, you can use the localStorage.getItem() method, passing the key of the item you want to retrieve.
- Removing Data: You can remove an item from localStorage using the localStorage.removeItem() method, passing the key of the item you want to remove.
- Clearing All Data: To remove all items stored in localStorage, you can use the localStorage.clear() method.

```
<!DOCTYPE html>
<html>
<head>
    <title>localStorage Example</title>
</head>
<body>
    <script>
        // Storing data in localStorage
        localStorage.setItem('username', 'john_doe');
        localStorage.setItem('theme', 'dark');

        // Retrieving data from localStorage
        const username = localStorage.getItem('username');
        const theme = localStorage.getItem('theme');

        console.log('Username:', username);
        console.log('Theme:', theme);

        // Removing data from localStorage
        localStorage.removeItem('theme');

        // Clearing all data from localStorage
        // localStorage.clear();
    </script>
</body>
</html
```

HTML 5 Session storage

- HTML5 has sessionStorage object, which is similar to localStorage, but the data stored in sessionStorage is cleared when the session ends (when the browser is closed or the tab is closed).
- The data is not persistent across multiple sessions like localStorage.

```html
<!DOCTYPE html>
<html>
<head>
    <title>sessionStorage Example</title>
</head>
<body>
    <script>
        // Storing data in sessionStorage
        sessionStorage.setItem('username', 'Ankit Patel');
        sessionStorage.setItem('theme', 'light');

        // Retrieving data from sessionStorage
        const username = sessionStorage.getItem('username');
        const theme = sessionStorage.getItem('theme');

        console.log('Username:', username);
        console.log('Theme:', theme);

        // Removing data from sessionStorage
        sessionStorage.removeItem('theme');

        // Clearing all data from sessionStorage
        // sessionStorage.clear();
    </script>
</body>
</html>
```