HTML 5

Created By :- The EasyLearn Academy

www.theeasylearnacademy.com

theeasylearn@gmail.com

Contact No :- 9662512857

What is HTML5?

- HTML5 is the new standard for HTML.
- The previous version of HTML, HTML 4.01, came in 1999.
- The web has changed a lot since after that.
- HTML5 is still a work in progress and it will completed by the 2022.
- However, the major browsers support many of the new HTML5 elements and APIs.

Browser Support for HTML5

- HTML5 is not yet an official standard, and no browsers have full HTML5 support.
- The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.
- The pre installed mobile web browsers in smart phone have excellent support for HTML5.

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

- Big
- Center
- tt
- **Font**
- **u**
- **Strike**
- basefont
- frame
- frameset
- noframes
- applet

The HTML5 <!DOCTYPE>

- In HTML5 there is only one <!doctype> declaration, and it is very simple:

  <!DOCTYPE html>
- The doctype appears just above the <html> tag, at the very start of each document you write:
- The doctype tells the browser how to render(process) the page.

Minimum HTML5 Document

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

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

<body>
</body>

</html>
```

HTML <article> Tag

- The article is used for part of a page that has individual topic/matter in a document, page, application, or site.
- it is independently distributable or reusable, e.g. in syndication.
- article can be a forum post, a magazine or newspaper article, a blog entry, a user-submitted comment, an interactive widget or gadget, or any other independent item of content.
- In addition to its content, an <article> element typically has a heading (often in a header element), and sometimes a footer.

<header> Tag

- The <header> tag specifies a header for a document or section.
- The <header> element should be used as a container for introductory content or set of navigational links.
- You can have several <header> elements in one document.
- A <header> tag cannot be placed within a <footer>, <address> or another <header> element.

<footer> Tag

- The <footer> tag defines a footer for a document or section.
- A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.
- You can have several <footer> elements in one document.
- Contact information inside a <footer> element should go inside 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
- It could also appear stand alone in its own right.
- 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 poings </li>
    - <li> secondkey poings </li>
  - </ol>
  - </aside>
  - </artical>
- </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+	NO	YES	YES
Safari 5+	YES	YES	NO
Opera 10+	NO	YES	YES

HTML <audio> Tag

- **MIME Types for Audio Formats**

Format	**MIME-type**
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

- **Example**
- Play a sound:

```
<audio controls>
   <source src="horse.ogg" type="audio/ogg">
   <source src="horse.mp3" type="audio/mpeg">
  <source src="horse.wav" type="audio/wav">
   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
- www.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 id="audioplayer" name="audioplayer" preload controls loop
    style="width:424px;">
   <source src="a1.mp3" type="audio/mpeg" />
   <source src="a1.ogg" type="audio/ogg" />
   <source src="a1.wav" type="audio/wav" />
    <embed src="a1.mp3" type="application/x-mplayer2" autostart="true"
    playcount="true" loop="true" height="0" width="0">
</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.
- Users will see a drop-down list of pre-defined options as they input data.
- Value of the <input> element's list attribute must match with id attribute of with a <datalist> element.
- The datalist tag is not supported in Internet Explorer 9 and earlier versions, and in Safari.

Example

```
<form action="demo_form.php" method="get">
<input list=“course" name=“txtcourse">
<datalist id=“course">
    <option value=“BCA">
    <option value=“MCA">
    <option value=“BBA">
    <option value=“B.Com">
    <option value=“Other">
</datalist>
<input type="submit">
</form>
```

<details> Tag

- The <details> tag specifies additional details that the user can view or hide on demand.
- It is like accordin control in jquery.
- The <details> tag can used to create an interactive widget that the user can open and close. Any sort of content can be put inside the <details> tag.
- The content of a <details> element should not be visible unless the open attribute is set.

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.
- The <figure> tag is supported in Internet Explorer 9, Firefox, Opera, Chrome, and Safari.
- It is mostly used with figcaption tag.

example

-
  ```
  <figure>
    <img src=“sample.jpg" alt=“image not
  avilable" width=“500" height=“400">
  </figure>
  ```

<figcaption> Tag

- The < figcaption **>** element is used to add a caption for the <figure> element.
- The <figcaption> element can be placed as the first or last child of the <figure> element.

example

- <figure>
  <img src=“sample.jpg" alt=“image not avilable" width=“500" height=“400">
- <figcaption>this is sample caption of image </figcaption>
  </figure>

<mark> Tag

- The <mark> tag defines marked text.
- It is used to highlight parts of your text.
- It highlight the 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 defines a set of navigation links.
- Not all links of a document must be in a <nav> element. The <nav> element is intended only for major block of navigation links.
- Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

Example

```
<nav>
  <a href=“one.html”>HTML</a> |
  <a href=“two.html">CSS</a> |
  <a href=“three.html">JavaScript</a> |
  <a href=“four.html">jQuery</a>
</nav>
```

<time> Tag

- the <time> tag defines either a time (24 hour clock), or a date in the Gregorian calendar.
- The <time> tag does not render as anything special in any of the major browsers.
- This element can be used as a way to encode dates and times in a machine-readable way.
- for example, user agents can offer to add birthday reminders or scheduled events to the user's calendar, and search engines can produce smarter search results.

example

- <p>bank open at <time>10:00</time> every morning. And close at <time>5:00 pm </time></p>

  <p>I have exam on <time datetime="2008-02-14“></p>
- This tag can be used with pubtime extra attributes which is used to indicate browser that it is publication date of article in which this tag appears
- If datatime attributes is omitted then user has to give date/time in tag itself.

<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">
    <source src="movie.ogg" type="video/ogg">
  <source src=“small.webm” type=“video/webm”>
  <embed src=“small.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.

Input Type: color

- The color type is used for input fields that should contain a color.
- **Example**
- Select your favorite color:
- <input type="color" name="favcolor">

Input Type: date

- The date type allows the user to select a date.
- **Example**
- Birthday:
- <input type="date" name="bday">

Input Type: datetime

- The datetime type allows the user to select a date and time (with time zone).
- example
  - Birthday (date and time):
  - <input type="datetime" name="bdaytime">

Input Type: datetime-local

- The datetime-local type allows the user to select a date and time (no time zone).
- example
- Birthday (date and time):
- <input type="datetime-local" name="bdaytime">

Input Type: email

- **Input Type: email**
- The email type is used for input fields that should contain an e-mail address.
- Example
  - E-mail:
  - <input type="email" name="email">
- Safari on iPhone recognizes the email type, and changes the on-screen keyboard to match it (adds @ and .com options).

Input Type: month

- The month type allows the user to select a month and year.
- Example
  - Birthday (month and year):
  - <input type="month" name="bdaymonth">

Input Type: number

- The number type is used for input fields that should contain a numeric value.
- You can also set restrictions on what numbers are accepted:
- Example
  - Quantity (between 1 and 5): <input type="number" name="quantity" min="1" max="5">
- Use the following attributes to specify restrictions:
  - max - specifies the maximum value allowed
  - min - specifies the minimum value allowed
  - step - specifies the legal number intervals
  - value - Specifies the default value

Input Type: range

- **Input Type: range**
- The range type is used for input fields that should contain a value from a range of numbers.
- You can also set restrictions on what numbers are accepted.
- Example
  - <input type="range" name="points" min="1" max="10">

Input Type: time

- The time type allows the user to select a time.
- Example
  - Select a time:
  - <input type="time" name="usr_time">

Input Type: url

- The url type is used for input fields that should contain a URL address.
- The value of the url field is automatically validated when the form is submitted.
- Example
  - Add your homepage:
  - <input type="url" name="homepage">

Input Type: week

- The week type allows the user to select a week and year.
- Example
  - select a week:
  - <input type="week" name="week_year">

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="demo_form.php">
      First name: <input type="text" name="fname"><br>
      Last name: <input type="text" name="lname"><br>
      <input type="submit" value="Submit"><br>
      <input type="submit" formaction="demo_admin.php"
      value="Submit as admin">
    </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}" title="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

http://www.iana.org/assignments/media-types/media-types.xhtml

Attribute Values

- audio/*
    - All sound files are accepted
- video/*
    - All video files are accepted
- image/*
    - All image files are accepted

For complete list of MIME type refer to following URL

http://www.sitepoint.com/web-foundations/mime-types-complete-list/

<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 usegetCurrentPosition() 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.

```
<!DOCTYPE html>
<html>
<body>
<p id="demo">Click the button to get your coordinates:</p>
<input onclick="getLocation()" type="button" value="get location" />
<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 5 Local Storage

- HTML 5 local storage allows web pages to store data locally within the user's browser.
- Earlier, this was done with cookies.
- We can still use cookies but local storage is more secure and faster.
- The data is not included with every server request, but used ONLY when asked for.
- It is also possible to store large amounts of data, without affecting the website's performance.
- The data is stored in name/value pairs, and a web page can only access data stored by itself.
- Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

Types of local storage object

- HTML local storage provides two objects for storing data on the client:
  - window.localStorage –
    - stores data with no expiration date.
    - The data will not be deleted when the browser is closed, and will be available the next day, week, or year.
  - code.sessionStorage
    - stores data for one session (data is lost when the tab is closed)
    - The sessionStorage object is equal to the localStorage object,
    - **except** that it stores the data for only one session. The data is deleted when the user closes the browser window.

- How to store value into local storage
  - localStorage.setItem("name",”ankit”);
- How to read value from local storage
  - localStorage.getItem("name");
- How to store value into session storage

  sessionStorage.variable-name=value;
- How to read value from local storage

sessionStorage.variable-name