Jump2Learn™
PUBLICATION

www.jump2learn.com

Concepts of Web Technology

Jump2Learn - The Online Learning Place

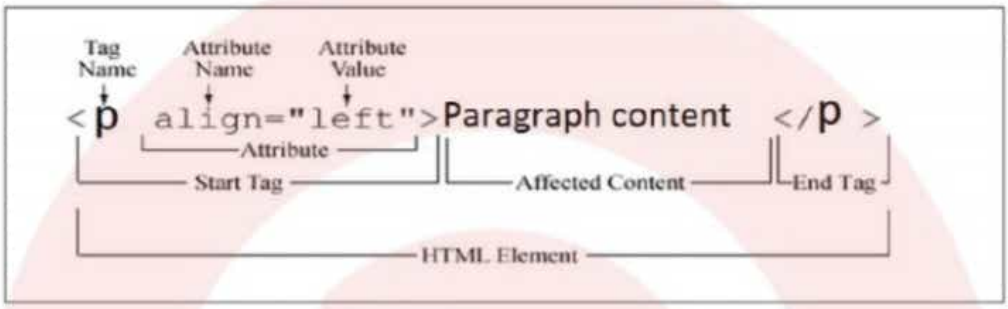
Dr. Jagin M. Patel | Ms. Meghna N. Vithlani

Unit- 1

Working With HTML5 And CSS

1.1 Introduction to HTML

- HTML stands for HyperText Markup Language. It is used to design web pages using markup. HTML is the combination of Hypertext and Markup language. Hypertext means the link between the web pages. Markups are the tags that define the structure of web documents. HTML elements are represented by tags. HTML tags label the content such as "paragraph", "heading", "table", etc. Browsers do not show the HTML tags but use them to display the content of the page.
- The structure of HTML elements is shown here:



- Tags are always embraced on either side by angle brackets. The letters inside the brackets indicate the type of tag it is. In the above example, *p* is the tag that refers to a paragraph in Html. The contents of the tag specify the content that the tag is structuring for a browser.
- Most tags have a closing tag. The closing tag is indicated by a forward slash (/) in front of the tag name inside of the angle brackets <>.
- Sometimes it is necessary to specify some additional content with a tag. It is called an attribute. In the above example *align* is the attribute and *left* is it's the value.

1.1.1 The Rules of HTML

- **Not Case Sensitive**

  These markup examples are all valid in html.

  <h1>Concept of web technology </h1>

  <H1> Concept of web technology </h1>

  <h1> Concept of web technology </H1>

  <H1> Concept of web technology </H1>

- **Sensitive to a single white space character**

  In Browsers, all-white spaces between characters are warped into a single space. This includes all tabs, line breaks, and carriage returns.

- **Follows a content model**

  HTML supports a strict content model. Certain elements suppose to appear within other elements.

Example:

```
<ul>
  <h1> Concept of web technology </h1>
</ul>
```

Here <h1> tag is not proper. The <ul> tag is supposed to contain <li> tags.

- **Elements have close tags except for the empty tag**

  Most of the elements have corresponding closing tags. However, it is optional. For example, both of the headings are allowed, but the first one is more preferable.

  <h1> Concept of web technology </h1>

  <h1> Concept of web technology

  Tags such as line break
, do not have closing tags. These tags are called empty elements.

- **Minimize unused elements**

  Sometimes tags may not give proper effect in a document. For example, the tag specifies a paragraph. When it used repeatedly, such as , it does not rendered as many blank lines because the browser reduce the empty elements.

- **Elements can be nested**

  Tags can be nested. For example

  Concept of web technology

- **Browsers do not consider unknown attributes and elements**

1.2 ABOUT HTML5

- HTML5 code will structure, format, and then work with a browser to present content to users. The HTML5 is the most current version of the HTML. It supports multimedia and technology.
- Most of the additional styling and design elements in web pages come from CSS or Cascading Style Sheet. CSS will be covered later in this chapter.
- The first step to learn HTML5 is to understand its building blocks. The basic building blocks of all HTML5 are elements, tags, and attributes.
- Most of the latest versions of browsers, such as Google Chrome, Apple Safari, Mozilla Firefox, and Opera support most of the HTML5 features. Internet Explorer 9.0 and higher versions also have support for HTML5 functionality.

- HTML5 is supported by the mobile web browsers that come pre-installed on iPads, iPhones, and Android phones.

1.2.1 Structure of HTML5 Document

- The structure of HTML documented pages contains:
  - A declaration at the top, which indicates that it is an HTML5 document
  - A document header
  - A document body
- An HTML5 document mainly consists of a Head and Body. The *head* contains document title, scripts, character set, styles, etc. The *Body* contains the actual content that web browsers display.
- **The following are found on every page.**

  (1). *<!DOCTYPE>* informs the browsers that it is an HTML 5 document.

  (2). *<html>:* The DOC Type Declaration is followed by *<html></html>* opening and closing tags. These tags contain Head, Body, etc inside the document.

  (3). <head>: After <html> opening, *<head> </head>* opening and closing tags come. These tags include information about the web page such as the title of the page, scripts, style, definitions, etc. Only certain markup elements can be used in the head. Some of these elements are listed below. These elements are known as HTML Head Elements. Meta elements are used to specify other metadata, such as page author, description, keywords, last modified, etc.

  The following is the list of different tags that can be used within the <head> tag.

Tag	Description
<head>	It specify information about the document
<title>	It specifies the title of a document
<base>	It specifies a default address or a default target for all links on a page
<link>	It specifies the relationship between a document and an external file
<meta>	It specify metadata about an HTML document
<script>	It specify a client-side script
<style>	It specify style information for a document

(4). *<body>*: After the closing head tag is the *<body> </body>* opening and closing body tags. The <body> element contains all the contents of an HTML document. It specifies the content to be displayed on the browser. This content includes hyperlinks, paragraphs, labels, headings, tables, images, etc. The <body> must appear after the <head> tag. In other word, it is placed between </head> and </html> tags. This is a necessary tag for every HTML document. It can be used only once in the entire document.

- The following figure shows the structure of an HTML5 document

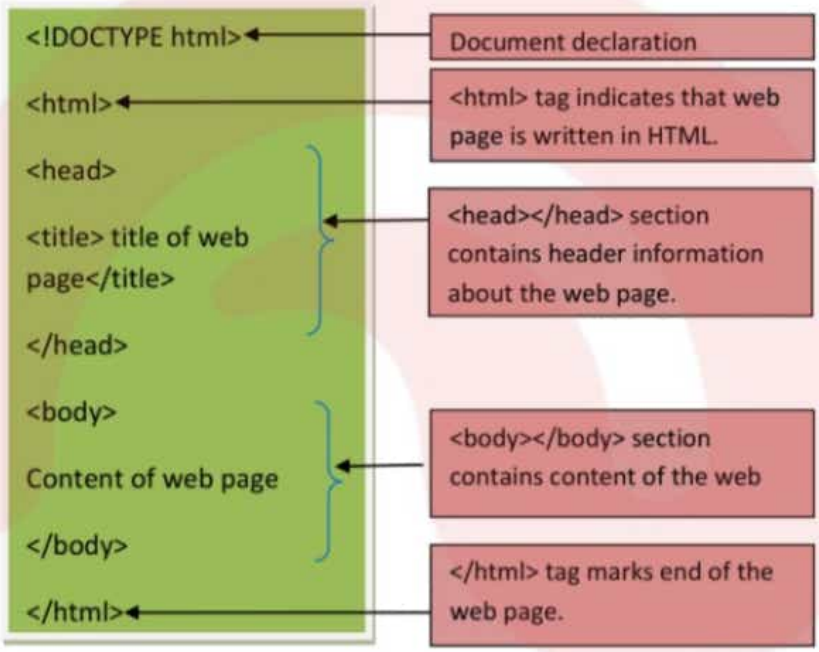


Figure 1.1 Structure of HTML5 Document

Example

```
<!DOCTYPE HTML>

<html>
<head>
    <meta charset="utf-8">
    <title>Basic Page Structure</title>
</head>
<body>
    <h1>Creating an HTML 5 Document</h1>
    <p>This is a paragraph </p>
</body>
</html>
```

1.2.2 Features of HTML5

Following is a set of some features introduced in HTML5.

- **New Semantic Elements** – New semantic tags such as <header>, <footer>, and <section> are introduce.
- **Forms 2.0** – HTML web forms are improved. New attributes are introduced for <input> tag.
- **WebSocket** – This feature provides two-way communication technology for web applications.
- **Drag and drop** –It is possible to Drag and drop the items from one place to another place within the webpage.
- **Server-Sent Events** – HTML5 introduces events called Server-Sent Events (SSE) which go from the webserver to the web browsers.
- **Canvas** – This feature allows two-dimensional drawing that can be programmed with JavaScript.
- **Audio & Video** – It is easy to embed audio and video on web pages without the need for third-party plugins.
- **Persistent Local Storage** – store data locally within the user's browser without the need for third-party plug-ins.
- **Geolocation** – It is possible to share physical location with the web application.
- **Microdata** – It allows to create own vocabularies and extend web pages with custom semantics.

1.2.3 Difference between HTML and HTML5

- There are many differenDces between HTML and HTML5:

FEATURES	HTML	HTML5
Multimedia support	It didn't support audio and video without third-party plug-ins.	It supports audio and video using <audio> and <video> tags.
Browser Support	It is supported by old browsers.	It is supported by all-new browsers such as Chrome, Firefox, Mozilla, Safari, etc.
Storage	It uses cookies to store temporary data.	It uses SQL databases and Web storage to store data.
JavaScript support	Does not allow JavaScript to run in the browser.	Allows JavaScript to run in the background using JavaScript Web worker API.

Vector and Graphics	Vector graphics is possible using technologies such as VML, Silver-light, Flash, etc.	Vector graphics is inbuilt and possible using SVG and canvas.
drag and drop effect	Do not allow drag and drop.	Allows drag and drop.
Shapes	It does not allow to draw shapes such as triangles, circles, etc.	It allows drawing various shapes such as triangles, circles, etc.
mobile-friendly	It is less mobile-friendly.	It is more mobile-friendly.
Doctype	Doctype declaration is too long and complicated.	Doctype declaration is simple and easy.
Semantic element	Semantic elements like nav, header was not present.	Semantic elements for web structure like nav, header, etc. are available.
Character Encoding	long and complicated.	simple and easy.
Geolocation	It is impossible to get the physical location of the user.	It is possible to get the location of the user using JavaScript GeoLocation API
Handling inaccurate syntax	It is not capable to handle inaccurate syntax.	capable of handling inaccurate syntax.

- **Many HTML elements have been changed or removed from HTML5. The following are a few of them:**

ELEMENT	IN HTML5
<applet>	Changed to <object>
<acronym>	Changed to <abbr>
<dir>	Changed to
<frameset>	Removed
<frame>	Removed
<noframes>	Removed
<strike>	No new tag. CSS is used for this
<big>	No new tag. CSS is used for this

<basefont>	No new tag. CSS is used for this
<font>	No new tag. CSS is used for this
	No new tag. CSS is used for this
<tt>	No new tag. CSS is used for this

- Many new elements are added in HTML5. A few of them are header, footer, nav, audio, video, figure, figcaption, progress, command, time, datalist, meter, data, section, aside, canvas, summary, rp, rt, details, wbr, keygen, embed, article, hgroup, bdi, mark, output, source, track, section, etc.

1.3 Concept of CSS

1.3.1 Introduction

Cascading Style Sheets, also known as CSS, is a simple design language. It simplifies the process of making web pages presentable.

CSS gives the look and style to web pages. Using CSS, we can deal with the style of fonts, text color, spacing between paragraphs, column sizing and layout, background images, background colors, layout designs, a variety of effects, variations in display for different devices, and screen sizes, and much more.

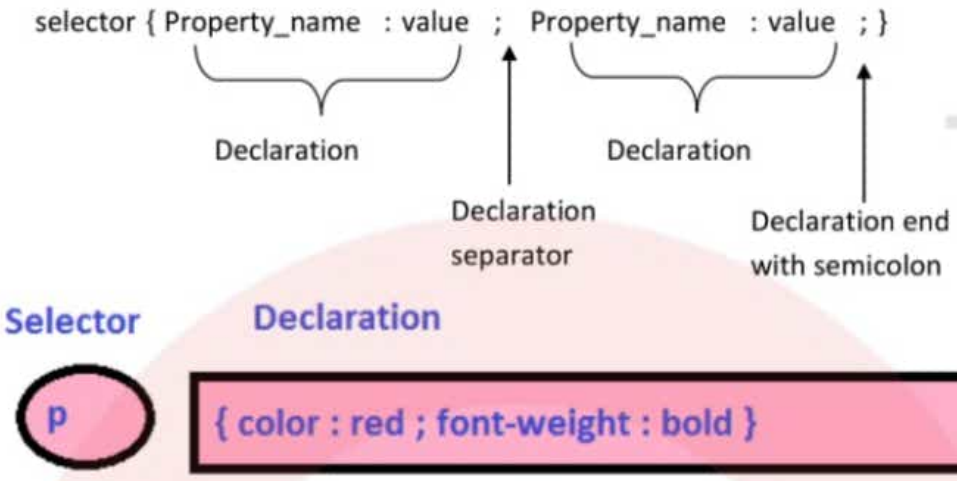
CSS is easy to learn and understand. It gives great control over the presentation of an HTML document. CSS is combined with the HTML or XHTML.

1.3.2 Advantages of CSS

- **CSS saves time:** CSS saves a lot of work. We can write CSS once and then reuse in various HTML pages.
- **Pages load faster**: It eliminates writing HTML tag attributes every time. Simply write CSS for tag and apply it to all the occurrences of that tag. The file size of the CSS is very small, So website load faster.
- **Simple to maintain:** To make a overall change, simply modify the style. This modified style will be updated automatically in all the web pages.
- **Better styles to HTML**: In comparison to HTML attributes, CSS support many attributes. So better look can be given to HTML page.
- **Device Compatibility**: Style sheets allow content to be optimized for multiple devices. Different versions of a website can be displayed for various devices such as PDAs, cell phones, or for printing, by using same HTML document.
- **Browser support**: All browsers support CSS. Now HTML attributes are being deprecated. So it's a better idea to use CSS.

1.3.3 CSS Syntax

The basic syntax is as follows:



A CSS is made of style rules. These rules are understand by the browser and then applied to the related elements in the document. A style rule is made of three parts:

1. **Selector** : A selector is an HTML tag on which a style to be applied. It can be any tag such as <p>, <h5> , <table> etc.
2. **Property name** : A property is type of attribute of HTML tag. All the HTML attributes are converted into CSS properties. It may be height, color, border etc.
3. **Value** : Values are given to properties. For example, *color* property can have value red. Property name and value separated by a colon (:).

There are various types of selector.

Selector	Description	Example

element selector	Select elements based on the element name.	h1 {
		text-align: center;
		color: red;
		}

id selector	define style rules based on the id attribute of the elements.
	To select an element with a specific id, use a hash (#) character before the element id.
		#div1 {
		text-align: center;
		color: red;
		}

class selector	Select elements with a specific class attribute.
	To select elements with a specific class, use a period (.) character before class name.
		.big {
		font-weight: bold;
		color: red;
		}

1.3.4 Applying CSS to a Document

Styling can be added to HTML elements in 3 ways:

1	Internal (Embedded)	document-wide style sheet specified by the <style> tag in <head>
2	Inline	using a style attribute in HTML elements
3	External /linked	using one or more external CSS files, file linked by a <link> tag

a) Document-wide style sheet (Internal Style Sheet)

- It defines a style for one HTML page
- It is defined with the <style> tag within the <head> element of an HTML document.

```
<style type="text/css">
    style rules here
</style>
```

- Within the style block, style sheet rules are included.
- **Example**:

Following example create CSS for <p>. This CSS is applied to all <p> tag. Here three rule are created for all <p> tag: (1) color is set red (2) left margin is set to 100px (3) font size is set to 36pt.

```
<html>
    <head>
      <style>
            p
        {
          color: red;
          margin-left: 100px;
          font-size: 36pt;
        }
      </style>
    </head>
    <body>
      <p> 1. css on paragraph</p>
      <p>2. css on paragraph </p>
    </body>
</html>
```

b) Using Inline Style

- Inline styling is used to apply a style to a single HTML element
- You can add style information directly in a single element.

- It is applied using *style* attribute. The *style* attribute can have any CSS property.
- To create inline styles, add the *style* attribute to the element.

Example:

The following example creates CSS for particular <p> tag. Here colour is set red, left margin is set to 100px, and font size is set to 36pt for particular <p>.

```
<p style="color: red; margin-left: 100px; font-size: 36pt; ">  1. css on heading1
</p>
```

- The main problem with this type of CSS is that inline rules are applied to a particular tag.
- If you want to give style to more than one <p>, it is compulsory to copy- paste the style attributes into every other <p> tag.
- Here, markup and CSS presentation is mixed up.

c) Linking to a Style Sheet (External Style sheet)

- An external style sheet is perfect when the style is applied to multiple pages.
- Using an external style sheet, it is possible to change the look of all Web pages by modifying one file.
- Each page must link to the style sheet using the <link> tag. The <link> tag is placed inside the head section:

```
<head>
   <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

- To create an external style sheet, you can use any text editor. The file should not contain any html tags. Style sheet file saved with a .css extension.
- **How to create and use external CSS**

Step -1: Create css in file with extention .css (say mystyle.css)

Step -2 : Use *<link>* in <head> element to link external css to webpage:

```
<link href="mystyle.css" rel="Stylesheet" type="text/css" />
```

Attribute	Description
href	It specifies the URL or location of the CSS file
rel	It is required attribute and set to "stylesheet". It specifies the relationship between the current document and the linked CSS file.
type	It specifies the media type of the linked file.

Step -3: set *Class* property of control on which you want to apply CSS.

Example:

- An example of a style sheet file "mystyle.css" is shown below

```
H1 { color:red; }
p { margin-left:50px; }
h2 { color:green;}
```

Comparison of Style Sheet Approaches			
	External Style Sheets	**Document-Wide Style**	**Inline Style**
Syntax	<link rel="stylesheet" href=" mystyle.css" />	<style type="text/css"> p {color: red;} </style>	paragraph
Advantages	It set style for multiple web pages with single style sheet. Style information cached by the browser.	It easily controls style for entire web page. No supplementary page require for style information.	It easily controls style to a single element. It overrides any external or document styles.
Disadvantages	It need more download time, so there may be delay in page loading.	Style must be reapplied for other documents	Style must be reapplied throughout the document and outside documents.

Cascading order

- What style will be used when there is more than one style specified for an HTML element?
- Style applied using following priority:
  1. Inline style
  2. Internal style sheet
  3. External style sheet
  4. Browser default
- An inline style has the highest priority. Inline style override a style defined inside the <head> tag, or in an external style sheet.

1.3.5 Some Basic CSS Properties

a) Color and Background Property

Property	Description
Color	It is used to sets the text color of an element.
background-color	It sets the background color of an element
background-image	It sets the background image of an element. Repeats an image both horizontally and vertically.
background-repeat	It specify how a background image is repeated
background-attachment	It specify whether a background image is fixed or scrolls with the rest of the page
background-position	It specify the starting position of a background image

b) Text Property

Property	Description
Color	It sets the color of a text
line-height	It sets the distance among lines
letter-spacing	It sets the space between characters
text-align	It aligns the text in an element
text-decoration	It decorate the text
text-indent	It Indents the first line of text in an element
text-transform	It controls the letters in an element

c) Font Property

Property	Description
Font	It sets all the font properties in one declaration.
font-family	It specifies the font family.
font-size	It specifies the font size
font-style	It specifies the font style
font-variant	It specifies whether text displayed in a small-caps font or not
font-weight	It specifies the weight of a font

d) List Property

Property	Description
list-style	It sets all the properties for a list in one declaration
list-style-image	It specifies an image as the list-item marker

list-style-position	It specifies where to place the list-item marker
list-style-type	It set the type of list-item marker

e) Margin Property

Property	Description
margin-left	It sets the left margin of an element
margin-right	It sets the right margin of an element
margin-bottom	It sets the bottom margin of an element
margin-top	It sets the top margin of an element

f) Border Property

Property	Description
Border	It sets all the border properties in one declaration
border-color	It sets the color of the all borders
border-style	It sets the style of the all borders
border-width	It sets the width of the all borders

1.4 HTML Style

- Setting the style of an HTML element, can be done with the *style* attribute.
- The HTML *style* attribute has the following syntax:

<*tagname* style="*property*:*value;*">

Example:

```
<body style="background-color: lightblue;color:white;">
```

1.5 Comments in HTML.

- The contents of HTML comments are not displayed within a browser. Comments are denoted by a start value of <!- - and an end value of - ->.
- Comments can be in multiple line.

Example:

```
<!- -
  This is a Comment.  Comment will not be displayed
-->
```

1.6 HTML Formatting

- HTML provides many tags to format text without using CSS.
- These formatting tags are used to make text bold, italic, underlined, etc.
- There are two categories of formatting text:
  - Physical tag: These tags are used to give the visual look to the text.
  - Logical tag: These tags are used to give logical or semantic value to the text.

a) and Elements

The HTML ** element defines bold text.
Example: This text is bold
Output: This text is bold.

The HTML ** element defines strong text, with added semantic "strong" importance.

Example: This text is strong
Output: This text is strong.

b) and Elements

The HTML ** element defines italic text, without any extra importance.

Example: This text is italic
Output: *This text is italic.*

The HTML ** element defines emphasized text, with added semantic importance.

Example: This text is emphasized
Output: *This text is emphasized*

c) <small> Element

The HTML *<small>* element defines smaller text:

Example: this is <small>Small</small> Formatting
Output: this is Small Formatting

d) <big> Element

The HTML *<big>* element defines bigger text:

Example: this is <big>Big Font</big> Formatting style
Output: this is Big Font Formatting style

e) <mark> Element

The HTML *<mark>* element defines marked or highlighted text:

Example: this is <mark>Marked</mark> Formatting
Output: this is Marked Formatting

f) <del> Element

The HTML *<del>* element defines ~~deleted~~ (removed) text.

Example: My favorite car is ~~wagon R~~ Baleno.
Output: My favorite car is ~~wagon R~~ Baleno.

g) <ins> Element

The HTML *<ins>* element defines inserted (added) text. This text display using underline.

Example: My favorite car is <ins>Baleno</ins>
Output: My favorite car is Baleno

h) <sub> Element

The HTML *<sub>* element defines subscripted text.

Example: The chemical formula of water is H ₂O.
Output: The chemical formula of water of is H_2O.

i) <sup> Element

The HTML *<sup>* element defines superscripted text.

Example: The fractions of (a+b)₂ are a₂ + 2ab + b₂.
Output: The fractions of $(a+b)^2$ are $a^2 + 2ab + b^2$.

Summary of HTML Formatting tags:

Tag	Description
<b>	Defines bold text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines smaller text
<big>	Defines bigger text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted or underline text
<del>	Defines deleted or strike out text
<mark>	Defines marked/highlighted text

1.7 HTML Heading

- A HTML heading is used to display title or a subtitle on the webpage.
- HTML supports six different levels of headings *<h1>* to *<h6>*.
- The highest level header format is *<h1>* (main heading) and the lowest is *<h6>* (least important heading).
- The size of the heading is depends on the level selected, i.e <h1> to <h6>.

Example:

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
```

Output:

This is heading 1
This is heading 2
This is heading 3
This is heading 4
This is heading 5
This is heading 6

1.8 HTML Paragraphs

- HTML paragraph or HTML *<p>* tag is used to define a paragraph in a webpage.
- A paragraph always starts on a new line. It indicates a block of text.
- When we add some extra space or line inside a paragraph with HTML, the browser wraps up to a single white space in display.

Example:

```
<p> This is first simple paragraph without any extra space or line breaks. </p>
<p> This    paragraph    has
                some extra spaces and
            line breaks. </p>
```

Output:

This is first simple paragraph without any extra space or line breaks.
This paragraph has some extra spaces and line breaks.

- As it is seen in output that the browser will automatically remove any extra spaces and lines when the page is displayed.

1.9 HTML Line Breaks, Horizontal Rule and Preformatted tags

- The HTML *
* tag defines a line break. It is an empty tag and it has no closing tag.
- The *<hr>* tag displays a thematic break or horizontal rule in HTML page. It is used to separate content in an HTML page.
- The HTML *<pre>* tag is used to give preformatted text. The text inside a <pre> element is displayed as it is in a fixed-width font (usually Courier). It preserves spaces as well as line breaks.

Example:

```
<p> This is a paragraph with <br>
      Line break</p>
<br>
<hr>
<pre>Poems or computer programs with indents
                  Can be formatted using
            Preformatted tag</pre>
```

Output:

This is a paragraph with
Line break.

Poems or computer programs with indents
Can be formatted using
Preformatted tag

1.10 HTML Lists

- In HTML there are three basic forms of lists:
  1. Ordered lists (<ol>)
  2. Unordered lists (<ul>)
  3. Definition lists (<dl>)

1.10.1 Ordered Lists

- An ordered list is enclosed by *
* and *
*. It defines a list in which order matters.
- Ordering typically is given by a numbering pattern, using Arabic numbers, letters, or Roman numerals.
- Ordered lists are more appropriate for creating simple outlines or step-by-step information. The list items are automatically numbered by the browser.
- The *type* attribute of <ol> is:
  - a for lowercase letters
  - A for uppercase letters
  - i for lowercase Roman numerals
  - I for uppercase Roman numerals
  - 1 for regular numerals (1 is the default value)

- The <ol> element also has a *start* attribute. This specifies starting value in numbering list.

Example:

```
<ol>
    <li> Chess </li>
    <li> Origami </li>
    <li> Cricket </li>
</ol>
```

Output:

1. Chess
2. Origami
3. Cricket

1.10.2 Unordered Lists

- An unordered list is define using *
* and *
*
- It displays unordered bulleted lists of items.
- Here, bullet may be a filled circle, a square, or an empty circle.
- Use *type* attribute to set the bullet type for a list.
- The possible values for *type* attributes are: disc, circle, square

Example:

```
<ul>
    <li> Chess </li>
    <li> Origami </li>
    <li> Cricket </li>
</ul>
```

Output:

- Chess
- Origami
- Cricket

1.10.3 Definition List

- This list is a list of elements in definition form such as found in dictionary.
- Definition is created using three tags:
    i. <dl> tag is used to defines the description list. i.e. lists are enclosed within <dl> and </dl>.
    ii. <dt> tag is used to defines data term.
    iii. <dd> tag is used to defines data definition or description.

Example:

```
<dl>
  <dt> Website: </dt>
  <dd> Website is a collection of web pages. </dd>
  <dt> Browser: </dt>
  <dd>It is application software used to display web pages.</dd>
</dl>
```

Output:

Website:
Website is a collection of web pages.
Browser:
It is application software used to display web pages.

1.11 HTML Links

- A webpage can contain various links. When these links are clicked, it takes you directly to other pages or specific parts of a given page. These links are known as hyperlinks.
- Hyperlinks allow the user to navigate between Web sites by clicking on images or text. So, it is possible to create hyperlinks using text or images available on a webpage.
- A link is specified using HTML tag *<a>*. This tag is called an anchor tag. Anything between the opening <a> tag and the closing ** tag will be a part of the link that a user can click to access the linked document.
- An anchor can include text, images, or a combination of the two.
- For linking purposes, the *href* attribute is used. The *href* attribute is set to the URL or address of the target resource or document.
- **Attributes :**
  - Use the *<a>* element to define a link
  - Use the *href* attribute to define the URL or link address
  - Use the *target* attribute to define where to open the document when the user clicks the link.
  - Use the *<img>* element (within <a>) to use an image as a link
  - Use the *id* attribute to define bookmarks on a page
  - Syntax:

     link text

Example:

Following example create a hyperlink on the text "click here". When the user clicks the text "click here", www.jump2learn.com will open.

```
<a href="https://www.jump2learn.com/"> Click here </a>  To visit
Jump2Learn Website
```

HTML Link Colors

- In all the browser, by default, a link will appear like this:
  - An unvisited link is underlined and blue
  - A visited link is underlined and purple
  - An active link is underlined and red
- It is possible to change the default colors of the link using CSS as shown below:

Example:

Following code displays unvisited link in red, visited link in yellow and active link in green color.

```
<style>
        a:link
        {
         color: red;
        }

        a:visited
        {
         color: yellow;
        }

        a:active
        {
         color: green;
        }
</style>
```

The target Attribute

- The target attribute indicates where to open the linked document when the user clicks the link.
- The target attribute can have one of the following values:

Value	Description
_self	This is the default value. It opens the linked document in the same window or tab.
_blank	It opens the linked document in a new window/tab
_parent	It opens the linked document in the parent frame
_top	It opens the linked document in the full body of the window

Example:

Following code display link document in the new tab when link is clicked.

```
<a href="https://www.jump2learn.com/"  target="_blank"> Click here </a>
To visit Jump2Learn Website
```

Link Titles

- The *title* attribute gives additional information regarding an element. The information is shown as a tooltip text when the user moves the mouse over the element.

HTML Links to a Page Section- a Bookmark

- HTML bookmarks are used to allow users to jump to particular parts of a Web page.
- The bookmarks can be useful if a webpage is very long.
- To make a bookmark, first, create the bookmark, and then add a link to it.
- When the user clicks the link, the page gives a scroll to the bookmark location.

Example:

- First, create a bookmark with the id attribute:
- <h2 id="C4"> Chapter 4 </h2>
- Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:
- Jump to Chapter 4
- Or, add a link to the bookmark ("Jump to Chapter 4"), from another page:
- Jump to Chapter 4

1.12 Absolute URL and Relative URL

- Every resource - such as HTML document, video, program, image etc. - available on the Web has an address. This address is a Uniform Resource Locator, or "URL".

- URLs typically consist of three parts:
  - The first part is the protocol which is used to access the resource.
  - The second part is a domain name. The name of the machine hosting the resource is called the domain name.

- The third part is the name of the resource itself. It is given as a path. It may be a filename or file path on the hosting machine.

- An absolute URL contains the entire address of the page or resources. It is the full URL of the page. It contains the protocol to the domain name (www.testing.com) and includes the resource location within the website in the folder system.

  Example:

  In following example, "https://www.jump2learn.com/webdesign.html" is the absolute URL. Here "https" is protocol, domain name is www.jump2learn.com, and "webdesign.html" is filename:

  ```
  <a href = " https://www.jump2learn.com/webdesign.html"> click here </a>
  ```

- The relative URL, does not use the full web address. The relative link is relative to the current page. It contains the location following the domain. It assumes that the link is on the same site and is part of the same root domain. Its path refers to a resource on the same machine as the current document. The relative path may use forward slash(/).

  Example:

  In following code "webdesign.html" is relative path. It searches for document "webdesign.html" in current folder.

  ```
  <a href = "webdesign.html"> click here </a>
  ```

  In following code, relative path point to folder one level up.

  ```
  <a href = "/webdesign.html"> click here </a>
  ```

- Web developer should estimate the base URL for resolving relative URLs according to the following precedence (highest priority to lowest):

  - The base URL is set by the BASE element.
  - The base URL is given by an HTTP header.
  - By default, the base URL is that of the current document.

- In HTML, URLs are important in many situations:
  - linking document or resource using the <a> and <link> elements.
  - linking an external style sheet using the <link> and <script> elements.
  - To add images in a page, in the <img> and <input> elements.
  - submitting form.
  - referring to metadata conventions describing a document in the element.

Difference between Absolute and Relative URL.

Absolute URL	Relative URL
URL contains more information	they are shorter and often more portable
It indicates exact location in file system. It is independent of current directory.	It indicates location of directory using current directory.
contains all the information necessary to locate a resource.	consists only of the path, and the resource
Format is: protocol://domain name /path/ resource	Format is: /resource
must use absolute URLs when referring to links on different servers.	Relative URLs can take a number of different forms.
include the protocol (e.g. http://) and the domain name.	use to reference links on the same server as the page

1.13 Images in HTML

- In HTML, images are defined with the <img> tag.
- The <img> tag is empty tag and does not have a closing tag
- It has attributes only.
- Following are the attributes of <img> tag:

Attribute	Description
src	It specifies path or URL of an image
alt	It specifies an alternate text for an image. If browser unable to display image for whatever reason, it display alternate text.
style	It is used to specify the width and height of an image.
height	It is used to specifies the height of an image
width	It is used to specifies the width of an image
ismap	It is used to specifies an image as a server-side image-map
usemap	It is used to specifies an image as a client-side image-map

Example:

Following code display image. Here image height=100 and width=100. If browser not found file jump2learn_logo.jpeg, it will display message ="image not available".

```
<img src="jump2learn_logo.jpeg" alt="Learning Website" height="100"
width="100" alt="image not available">
```

This code can be written using style attribute:

```
style="width:100px;height:100px;"

<img src="jump2learn_logo.jpeg" alt="Learning
Website" style="width:100px;height:100px;"
 alt="image not available">
```

Output:

We can also link an image with other page. i.e. We can use an image as a link. To do this, put <img> tag inside the <a> tag.

Example:

```
<a href="https://www.jump2learn.com">
<img src="jump2learn_logo.jpeg"  height="100" width="100">
</a>
```

1.14 Tables In HTML

- Table is collection of row and column. Single row contain many columns.
- Intersection of row and column is called cell.
- HTML tables allow to arrange data into rows and columns.
- We can create a table to display data in tabular form. Table is created using <table>, <tr>, <td>, and <th> elements.
- HTML tables are also used to manage the layout of the page.
- Following is the list of tag used to create table.

Tag	Description
<table>	It defines a table.
<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.
<caption>	It defines the table caption.
<colgroup>	It specifies a group of one or more columns in a table.
<col>	It is used with <colgroup> element to specify column properties for each column.
<tbody>	It is used to group the body content in a table.
<thead>	It is used to group the header content in a table.

<tfooter>	It is used to group the footer content in a table.

- **colspan attribute:** It is used to make a cell span more than one column. It merge cells of a column.
- **rowspan attribute:** It is used to make a cell span more than one row. It merges cells of a row.

Example: simple table

Following code create simple table.

```
<html>
<body>
   <table>
    <tr>
      <th>NO</th>
      <th>NAME</th>
      <th>MARKS</th>
    </tr>
    <tr>
      <td> 1</td>
      <td> Manya</td>
      <td> 80</td>
    </tr>
    <tr>
      <td>2</td>
      <td> Navya</td>
      <td> 90</td>
    </tr>
        </table>
    </body>
    </html>
```

Output:

NO	NAME	MARKS
1	Manya	80
2	Navya	90

Example: table border

In above code add following lines to give border to table.

```
<style>
```

```
    table, th, td
    {
      border: 1px solid ;
    }
  </style>
```

Output:

NO	NAME	MARKS
1	Manya	80
2	Navya	90

Example: colspan

Following HTML code demonstrate colspan attribute. This code merges 2 columns in first row.

```
<html>
  <head>
    <style>
    table, th, td
    {
      border: 1px solid ;
    }
  </style>
</head>

<body>
  <table>
    <tr>
      <th  colspan=2>Subjects</th>
    </tr>
    <tr>
    <td>
      Subject 1
    </td>
    <td>
          Subject 2
        </td>
        </tr>
      </table>
    </body>
    </html>
```

Output:

Subjects	
Subject 1	Subject 2

Example: use of some more property

Following code create complex table using various tables tag and attributes.

```
<html>
  <head>

   <style>
     table, th, td
      {
       border: 1px solid black;
      }
     th, td
   {
      padding: 10px;
    }
   </style>
 </head>
 <body>
    <table>
     <caption>Result</caption>
     <thead>
      <tr>
        <th rowspan="2">ROLLNO</th>
        <th rowspan="2">NAME</th>
        <th colspan="3">SUBJECTS</th>
      </tr>
      <tr>
        <th>Subject1</td>
        <th>Subject2</td>
        <th>Subject3</td>
      </tr>
      </thead>
      <tbody>
      <tr>
        <td> 1</td>
        <td> Manya</td>
        <td> 100</td>
        <td> 94</td>
        <td> 91</td>
```

```
    </tr>
     <tr>
       <td>2</td>
       <td> Navya</td>
       <td> 80</td>
       <td> 88</td>
       <td> 75</td>
     </tr>
    </tbody>
    </table>
  </body>
</html>
```

Output:

Result

ROLLNO	NAME	SUBJECTS		
		Subject1	Subject2	Subject3
1	Manya	100	94	91
2	Navya	80	88	75

1.15 Forms in HTML

- HTML Forms are needed to collect some data from the user. For example, during user registration it is necessary to collect information like name, email address, age, etc.
- A form collect input from the user and then posts it to a back-end application which could be some server side scripts such as CGI, ASP Script or PHP script etc. The back-end application process data.
- The *<form>* tag is used to create an HTML form
- Syntax of <form>:

```
<form action = "URL" method = "GET|POST">
        form elements such as <input>
</form>
```

1.15.1 Form Attributes:

- Following is the list of attributes of <form> tag:

Attribute	Description
action	It specifies an address or URL where to submit the form. By default, form is submitted to page itself.
method	It specifies the method used while submitting the form. There are two possible values: GET or POST. The default value is GET.

name	It is used to identify the form
target	It specifies where to display the response that is received after submitting the form. It's possible values are: _blank, _self, _parent etc. It's default value is _self
novalidate	It specifies that form data is not validated when submitted.
rel	It specifies the relationship between a linked resource and the current document.
autocomplete	It specifies whether a form have autocomplete on or off. By default it is on.
enctype	It specifies how the form-data are encoded while submitting to the server. This attribute is applicable for method="post".

GET & POST METHOD OF FORM SUBMISSION:

- The default method of form data submission is GET.
- The form data is appended into the URL in the structure of name/value pair
- When GET is used, the submitted form data is visible in the page address field.
- If the form data have sensitive or personal information, then use POST.
- When POST method is used, the submitted form data is not visible in the page address field.

Difference between GET & POST Method

Factors	GET	POST
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
History	Parameters remain in browser history	Parameters are not saved in browser history
When preferred to use	Preferred for non-secure data, such as query strings in Google	Preferred for secure data
Security	Less secure because data sent is part of the URL.	More secure because the parameters are not stored in browser history
Data length limitation	Since this method adds the data to the URL and length of a URL is limited	No restrictions in sending data
Restrictions on data type	Only ASCII characters allowed	No restrictions on content type. Binary data is also allowed
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

1.15.2 Forms Elements:

- Following is the list of the HTML <form> element.
    - <label>
    - <input>
    - <select>
    - <textarea>
    - <button>
    - <datalist>

a) Form Input Elements

- <input> is most used element.
- It is used to create various interactive controls for forms to take data from the user.
- The "type" attribute of input element is used to create various types of controls. For example, <input type="text" name="name"> give in a text box.
- Following is a list of all types of <input> element of HTML.

Type	Description
text	It creates a one-line text input field
password	It creates password input field
submit	It creates a submit button to submit the form to server
reset	It creates a reset button to reset all values in the form.
radio	It creates a radio button. It allows to select one option.
checkbox	It creates checkboxes. It allows to select multiple options.
button	It creates a simple push button.
file	It creates controls from which user select the file from device storage.
image	It creates a graphical submit button.

- HTML5 added new types on <input> element. Following is the list of types of elements of HTML5

Type	Description
color	It is used to define an input field for color selection.
date	It is used to define an input field for date selection.
datetime-local	It is used to define an input field for entering a date without time zone.
week	It is used to enter the date with week-year, without time zone.
month	It is used to define an input field for taking a month and year, without time zone.
number	It is used to take a number in input field.
email	It is used to define an input field for taking an email address.
url	It is used to take URL
search	It is used to define a single line text field for entering a search string.
tel	It is used to enter telephone number.

Input Restrictions

- Here is a list of attributes that input restrictions :

Attribute	Description
disabled	It specifies that an input field is disabled
max	It specifies the maximum value for an inputted value
min	It specifies the minimum value for an inputted value
maxlength	It specifies the maximum number of characters that an input field can take.
pattern	It is used to specify a pattern or regular expression to check the input value.
readonly	It is used to specify that an input field is read-only i.e. value cannot be changed
required	It is used to specify that an input field must be filled out, i.e. input is required.
step	It is used to specify the number intervals for an input field
value	It is used to specify the default value for an input field
placeholder	It is used to specifies a short hint or clue regarding the input field

b) Checkbox Control

- Checkboxes are used when one or more option or choice is required to be selected.
- Checkbox is created using <input> tag with *type* attribute set to *checkbox*.
- When it is selected checkmark  will appear.
- Following is the list of attributes for the *<checkbox>* tag

Attribute	Description
Type="checkbox"	It Indicates that input control is a *checkbox*.
name	It is used to give a name to the control. It is used to refer to the form data after submitting the form or to refer to the element using JavaScript..
value	It is used to specify the value of the checkbox.
checked	It indicates that the check box is selected by default.

Example:

Following code creates a checkbox control with name="hobby" and value="cricket".

```
<input type="checkbox" name="hobby" value="Cricket">
```

c) Radio Button Control

- Radio buttons are used when out of many options, is required to be selected.
- They are created using *<input>* tag by setting *type="radio"*
- Radio buttons are used in MCQ type questions, gender selection, etc.
- Following is the list of attributes for radio buttons

Attribute	Description
type="radio"	It Indicates that input control is a *radio* button
name	It is used to give a name to the control. It is used to refer to the form data after submitting the form or to refer to the element using JavaScript.
value	The value that will be used if the radio button is selected.
checked	It indicates that the radio button is selected by default.

Example

Following code create radio buttons .

```
<input type="radio" name="status" value="m">Married
<input type="radio" name="status" value="f">Unmarried
```

Output:

○ Married ○ Unmarried

d) Textarea Control

- This is used when the user is required to give details that need multiple lines.
- Multi-line input controls are created using HTML *<textarea>* tag.
- It is useful to write a comment, a review, or a feedback form.
- Following is the list of attributes for the <textarea> tag

Attribute	Description
name	It is used to give a name to the control. It is used to refer to the form data after submitting the form or to refer to the element using JavaScript.
rows	It specifies the number of rows of the textarea box.
cols	It specifies the number of columns of the text area box.

- **Example:**

  Following code create textarea control name "review" having 6 rows and column width is 30.

```
<textarea name="review" rows="6" cols="30">
</textarea>
```

e) Label Control

- The *<label>* tag is used to define a label for a form control like text, textarea, etc.

Attribute	Description
form	It specifies one or more forms the label belongs to
for	It specifies which input control is bounded to this label. This value must be the same as the value in the input control's "id" attribute.

Example:

Following code create label that for text box with id="roll_no".

```
<label for="roll_no"> ROll No: </label>
<input type="text" id="roll_no">
```

f) Select Box Control

- A select box is also called the drop-down box.
- It is used to provide multiple options from where a user can select one or more options.
- It displays a list in the form of a drop-down list.
- One of the uses of the selection box is to provide a selection of city, state, country, etc on the registration page.
- It is created using *<select>* tag. The tag should contain one or more option elements. Each *<option>* tag specifies a menu choice.

- Following is the list of attributes for <select> tag

Attribute	Description
name	It is used to give a name to the control. It is used to refer to the form---data after submitting the form or to refer to an element using JavaScript
size	It indicates how many options visible in the drop-down at a time. If size > 1, then, It displays a scrolling list box.
multiple	It allows the user to select multiple items from the list.

- Following is the list of attributes of the <option> tag :

Attribute	Description
value	It is used to specify the value for a particular option.
selected	It is used to specify that the initial option is selected when the page loads.
label	An alternative way of labeling an option.

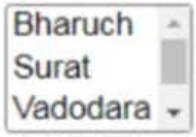
Example:

Following code displays a drop-down list with four options:

```
<select name="city" id="city">
        <option value="Bharuch">Bharuch</option>
        <option value="Surat">Surat</option>
        <option value=" Vadodara">Vadodara</option>
        <option value="Valsad">Valsad</option>
</select>
```

Output:

In previous example if *size* = 3 is set, i.e. <select size=3 name="city" id="city">, then output will look like this:



g) Button Controls

- Button control is used to create clickable buttons.
- You can create a clickable button using *<input>* tag by setting its type attribute to *button*.
- It is possible to put text and tags such as <img>,
 , etc. inside <button>.
- Following is the list of attributes of the button

Attribute	Description
Type	It specifies the type of button.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	It is used to specify the value for button control.

- The possible value for the *type* attribute are:

Value of type attribute	Description
button	It creates a button that is used to trigger a client-side script when the user clicks the button.
submit	It is used to create a button that automatically submits a form. This is the default value.
reset	It is used to create a button that automatically resets form controls to their initial values.

Example: Reset Form using Button

Following code create reset button using *<button>*. To see the effect, first enter text in text box. Then click reset button.

```
<form>
        Enter no :< input type="text"/>
        <button type="reset">reset</button>
</form>
```

Output:

Enter no:
reset

onclick() event:

The *onclick* event occurs when the element is clicked. This event is used to execute JavaScript code when the user clicks the button.

Note: JavaScript is covered in chapter 3.

Syntax:

```
<Button type="button"  onclick="JavaScript">
```

OR

```
<Button type="button"  onclick="function()">
```

Here function() is the name of the JavasScript function to be called when the user clicks the button.

Example: Calling JavaScript *onclick*

Following code call JavaScript function on button click. When button is clicked, JavaScript function and display message in alert box.

```
<script>
        function display()
        {
            alert("demo of button");
        }
</script>
<button type="button" onclick="display()"> display </button>
```

Difference between button created using <button> vs <input type="button" />:

- The <button> tag uses opening and closing bracket, while <input type="button"> uses only single tag.
- <button> can contain HTML. <input type="button" > is an empty element, therefore it cannot contain content.
- It is possible to put text and tags such as <img>,
 , etc. inside <button>. This is not possible with a button created using the <input> element.
- By default, <button> tag has the default attribute type="submit". If you don't specify the type attribute, then clicking the button will submit the form. If you want to achieve this behavior using the <input> tag, then set type="submit".

Tips:

The <button> is more semantic than the <input> with type of button (<input type= "button"). It is good idea to use the <button> element to make a clickable button.

h) Datalist Control

- The *<datalist>* element is used to give a list of permissible or pre-defined options for an <input> element.
- The *<option>* element is used to define list.
- As a user enters data, they will see a drop-down list of the pre-defined options related to entered data.
- It is used to give an auto-complete feature on the form element.
- To use <datalist>, set the *list* attribute of the <input> element. The value of *list* attributes must refer to the "id" attribute of the <datalist> element.

Example:

Following code create city datalist.

```
<input list="city">
    <datalist id="city">
        <option value="Bharuch">
        <option value="Surat">
         <option value="Vadodara">
        <option value="Valsad">
        <option value="Ankleshwar">
    </datalist>
```

Output:

After pressing "v", datalist will look like:

Example: Form creation using various control

```
<html>
  <head>
    <title>Registration Form</title>
  </head>
  <body>
    <h1> REGISTRATION FORM</h1>
    <form name="f1" action="" method="post">
      <table>
        <tr>
          <td><label> ENTER YOUR NAME: </label></td>
          <td><input type="text" placeholder="enter name" name="t1"
                required></td>
        </tr>
        <tr>
          <td><label>ENTER PASSWORD:</label></td>
          <td><input type="password" name="p1" placeholder="enter
              password" required></td>
        </tr>
        <tr>
          <td><label>ENTER ADDRESS:</label></td>
          <td><textarea rows="3" cols="30" placeholder="enter permanent
              address"></textarea></td>
        </tr>
        <tr>
          <td><label> GENDER:</label></td>
```

```
      <td>
        <input type="radio" name="g" value="m">MALE
        <input type="radio" name="g" value="f">FEMALE
      </td>
    </tr>
    <tr>
      <td><label> HOBBIES</label></td>
      <td>
        <input type="checkbox" checked name="ch1" value="art" >ARTS
        <input type="checkbox" name="ch2" value="sport" >SPORTS
        <input type="checkbox" name="ch3" value="travel" >TRAVEL
        <input type="checkbox" name="ch4" value="comp">COMPUTERS
      </td>
    </tr>
    <tr>
      <td><label> QUALIFICATION:</label></td>
      <td>
        <select name="qualification">
          <option> SELECT </option>
          <option> SSc. </option>
          <option> HSc. </option>
          <option> Diploma </option>
          <option> Bachelors</option>
          <option> PG </option>
          <option> Others </option>
        </select>
      </td>
    </tr>
    <tr>
      <td><input type="submit" name="s" value="REGISTER"></td>
      <td><input type="reset" name="r" value="CLEAR"></td>
    </tr>
  </table>
 </form>
</body>
</html>
```

Output:

1.16 Media elements: Audio and Video In HTML

- Before HTML5, there was no standard for showing videos on a web page. It required a third-party plugin such as flash to play audios and videos on a web page. Now with HTML5, we don't require any third-party plug-in.
- HTML5 includes two new Media Tags: *<audio>* and *<video>*. These tags are used to add media content on a webpage.
- With the introduction of Audio and Video tags, sharing multimedia becomes an integral part of the web.

1.16.1 HTML5 Audio

- HTML5 defines a standard for the playback of audio files.
- The HTML5 *<audio>* element is used to embed audio in a web page. It is easy to add audio to a website.
- It supports native audio without the use of any third-party plug-in.
- It allows the user to play, pause, and auto-play the audio.
- Is support .mp3, .ogg and .wav files.
- <Audio> support following media types

File Format	Media Type
MP3	audio/mpeg
Ogg	audio/ogg.
Wav	audio/wav

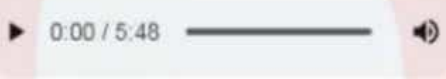
- Following is the list of attributes of the <audio> tag. These attributes help the user to control how the audio will play.

Attribute	Description
Src	It specifies the path or URL to the audio file.
Autoplay	It indicates that audio automatically begins to play.
Controls	It allows the user to control audio playback. Users have control over volume, seeking, and pause/resume playback.
Loop	It indicates that the audio automatically starts playing from the beginning after reaching the end. i.e. Means audio will start over again, every time after it is finished.
Type	Specifies the type of audio.

Example: embed audio

```
<audio src="sound.mp3" controls  type= "audio/mpeg ">
</audio>
```

Output:



Example: starts playing audio on the page load

Following cod start playing audio automatically by adding the *autoplay* attribute.

```
<audio src="sound.mp3" controls  type= "audio/mpeg "  autoplay>
</audio>
```

1.16.2 HTML5 Video

- HTML5 defines a standard for the playback of video files.
- The HTML5 *<video>* element is used to embed video in a web page. It is easy to show a video to a website using this element.
- It supports playing video without the use of any third-party plug-in.
- It allows the user to control over volume, play and pause the video.
- It supports .mov and H.264 files.
- Following is the list of attributes of <video> tag. These attributes help users to control how the video will play.

Attribute	Description
Src	It specifies the path or URL to the video file.
Controls	It allows the user to control video playback. The user has control over playing volume, seeking, and pause/resume playback.
autoplay	It indicates that the video automatically begins to play. i.e. play video on page load.
Loop	It indicates that the video automatically starts playing from the beginning after reaching the end. i.e. play video continuously even after it ends.
Width	It is used to define the width of the video player.
Height	It is used to define the height of the video player.
Poster	It is used to display an image on page load or until the user hits the play button. This property will not work if *autoplay* is set.
Audio	audio="muted" attribute will mute the audio of a video.
Track	It is used to add subtitles to the video

HTML Video support following media types:

File Format	Media Type
MP4 (MPEG4)	video/mp4
WebM	video/webm
Ogg	video/ogg

Example:

```
<video src="video.mp4" height=200 width=200 controls>
</video>
```


****