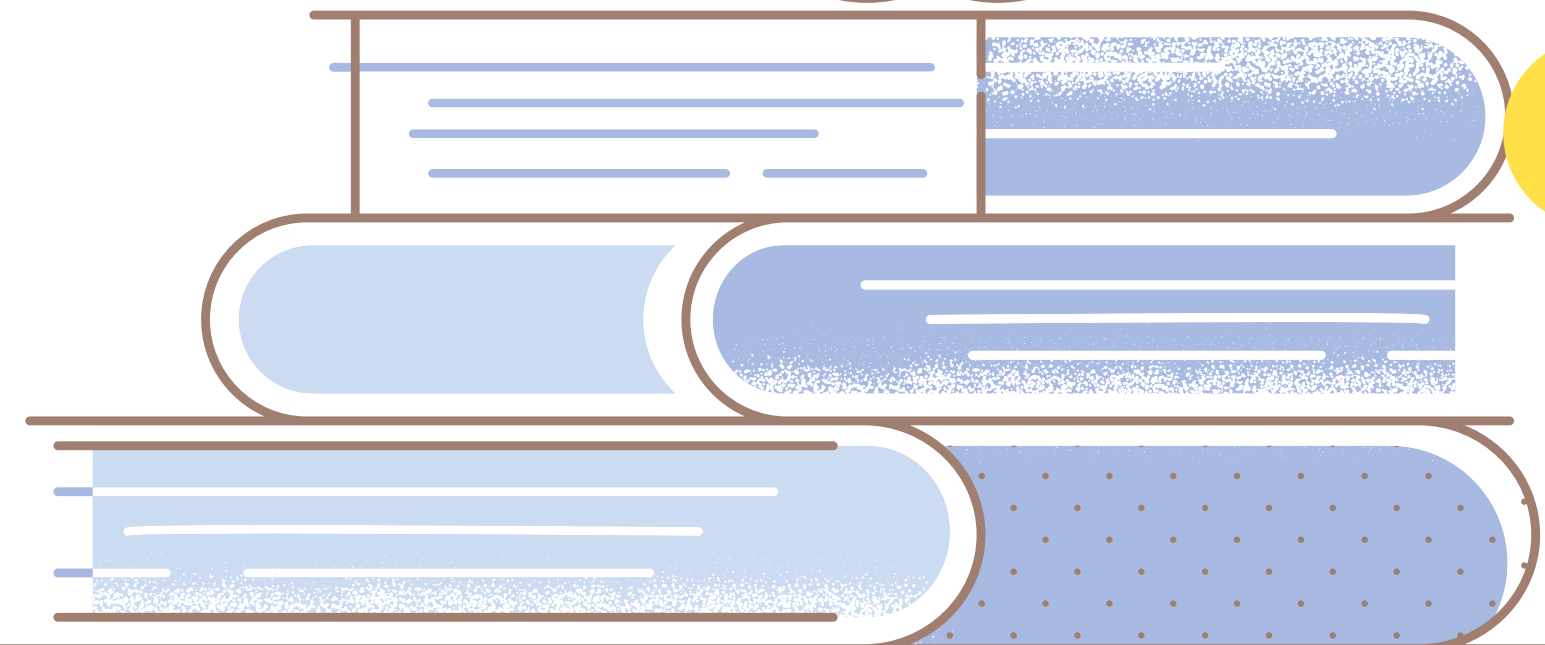# HTML

Checkout My Other Courses

Web Design Layouts A Must to Learn For All Levels - **Click Here**

HTML 5 from Beginner to Pro-Code [7 Days Challange] - **Click Here**

Javascript Complete Real time - Essential Course 2020 - **Click Here**

CSS Animation With Latest Effects - 2020 - **Click Here**

Beautiful Designs on SignUp , Contact , Login Forms - **Click Here**

Udemy

SPECIAL OFFER

SALE

75% OFF

SALE UP TO 75%

50% OFF

About book author

This is an unofficial free book created for educational purposes and is not affiliated with official HTML5 group(s) or company(s). Alltrademarks and registered trademarks are the property of their respective company owners

Learners Hub

Creative Director
&
Full Stack Developer

Limit of Liability/Disclaimer of Warranty:

The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This **Ebook** is **created free for education purpose** with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

Thank You...

The primary goal is to provide the best source of knowledge for you no matter what your specific skill level is. Although you can certainly use the book as a reference for specific topic. Some of the example through out the book do build on each other. By the end of the book, you learnt how to work with all the technology needed to create a master of this course..

What is HTML ?

HTML stands for **HyperText Markup Language.**

- First developed by **Tim Berners-Lee** in 1990.
- HTML is a Client Side Language.
- HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web
- HTML is not a Programming Language it is a Marking Langauge
- A MarkUp Language is a SET of Marking Tags
- HTML Uses a MarkUp Tags to describe a Web Page or Controls its Content.
- HTML is widely used language on the web.
- Using HTML We may just create a Static Website.

Why HTML Editor ?

The very fundamentals of HTML editors are same. They help you write code by highlighting syntaxes, insert commonly used HTML elements and structures as well as providing autocompletion.

Basically for HTML you may use Any Editor available in market…
Notepad which is by default available in all OS.

Advance Editor to Choose From

- Notepad++ (www.notepad-plus-plus.org)
- Atom (www.atom.io)
- Adobe Dreamweaver CC (www.adobe.com)
- Sublime Text (www.sublimetext.com)
- Visual Studio Code (www.code.visualstudio.com)

Learner's Hub
(B Praveen Kumar)

Installing Visual Studio Editor ...

First go to : **https://code.visualstudio.com/**

Based on the OS. Download the Software .... me on Windows

After Installation done successfully.. We will add Settings & Extensions based on our requirements.

VC - Setting	Visual Studio Description
Fonts	Font-size : 24 Editor : Font-Family : courier new

Learner's Hub
(B Praveen Kumar)

Visual Studio	Visual Studio Extension
HTML CSS Extension	Auto Close Tag
	Auto Rename Tag
	CSS Peek
	CSS-auto-prefix
	Google Fonts
	HTML Snippets
	html tag wrapper
	IntelliSense for CSS Class names in HTML
	Live Server
	Notepad++ Keymap
	Tag Inserter

Learner's Hub
(B Praveen Kumar)

Visual Studio	Visual Studio Extension
VC Themes	Material Theme Sublime Material Theme **Dracula at night** ✅ Monokai

Learner's Hub
(B Praveen Kumar)

Basic Structure of HTML Document

```
<!DOCTYPE html>
<html>
  <head>

    <title> , <style> , <script> , <meta>

  </head>
  <body>

      // Visibility to Users on the Browser Screen

  </body>
</html>
```

Learner's Hub
(B Praveen Kumar)

Understanding Indepth...

<!DOCTYPE> : It lets the browser know how the document should be interpreted, by indicating what version or standard of HTML (or other markup language) is being used.

<html> : Holds the Complete Web Page Document

<head> : Allows us to hold Head Section of the Document.. Which Holds

- **<title>** Used to display Content on the Menu tab of the Browser
- **<style>** Allows us to Add CSS On the Web Document
- **<script>** Allows us to Add JavaScript on the Web Document
- **<meta>** Usually Done by SEO (Search Engine Optimization)

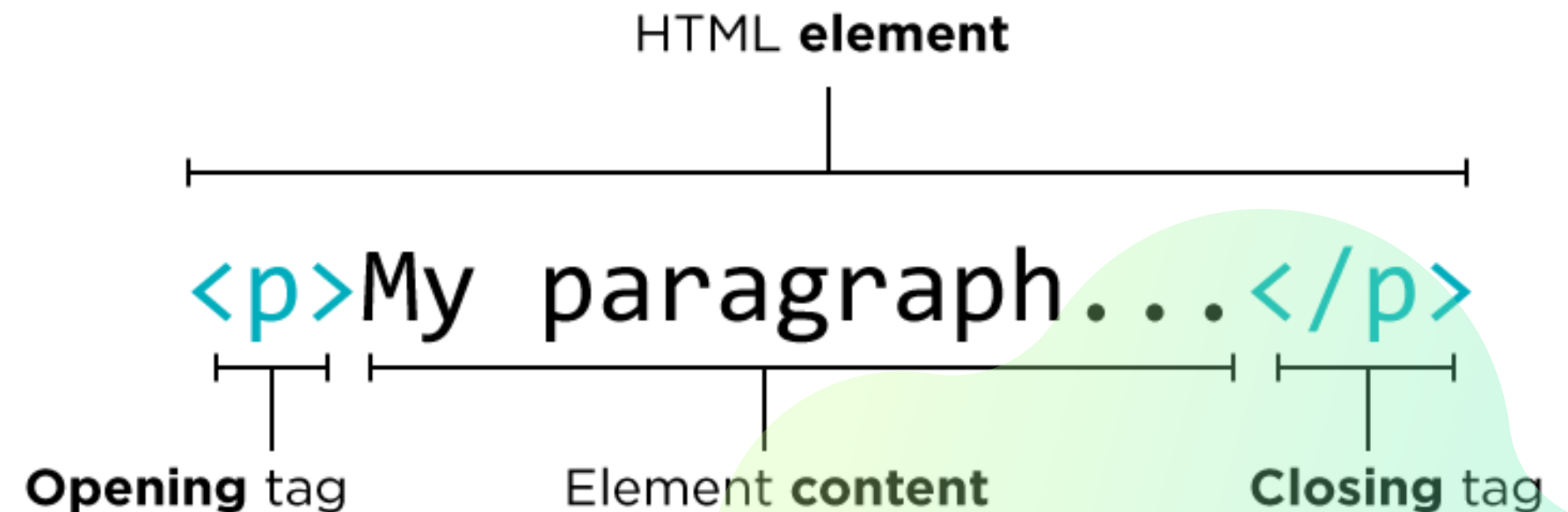**<body>** : it is Used to Display Content on the **Browser Screen**

How to Save and Execute the HTML Files

- All the HTML Files Should be Saved with **.html** as its Extension
- And this files can be Saved anywhere on your Computer
- And to Execute this Files we need the Browsers (Eg : IE , Chrome , Mozilla , Etc)

Learner's Hub
(B Praveen Kumar)

HTML Tags

- HTML Tags Comes with a Collection of **Pair Tags**
- Every Tag has Got its own Marking
- And this tags are case **Insensitive**
- HTML Tags must be Embeded in between **Angular Brackets**.

HTML **element**

`<p>`My paragraph...`</p>`

Opening tag   Element **content**   **Closing** tag

Elements in HTML

A tag which has Pair in them and also carry **Content** or **Graphical Display** between them then this are treated as Elements.

<p> Some Para Content.... **</p>**
**** Some Bold Content.... ****
<video src='path of video'**> </video>**

Non-Elements in HTML

A Tag With **No Pair** & **No Content** then this are treated as Non-Elements

**
** For Break Line
<hr> For Horizontal Line

Learner's Hub
(B Praveen Kumar)

Attributes in HTML

Attribute are Special Features which are supplied inside the HTML Elements and will appears within the html starting Tag...

Attribute Name   **Attribute Value**

`<p align = "right"> This is Some Para Content in html </p>`

Comments in HTML   `<!--   .......   -->`

Comments are mainly used not to display Content on top of the Browser.

`<!-- Inbetween this is ignored by the browser. -->`

Learner's Hub
(B Praveen Kumar)

HTML Formatting Tags

HTML provides several tags that you can use to make some text on your web pages to appear differently than normal text, for example, you can use the tag **<b>** to make the text bold, tag **<i>** to make the text italic and For more of Examples please watch Video..

List Tags in HTML

Grouping of Elements are Considered as List. HTML Supports 4 Types of List Tags.

- Ordered List
- Un-Ordered List
- Definition List
- Nested List

Ordered List in HTML

Grouping of Elements in the form of <u>Numbers , Alphabets and Roman Numbers</u> are Considered as **Ordered List**

<ol> Tag is Used as Ordered List
<li> Tag is the Child tag For <u>Ordered List</u>

ATTRIBUTES	VALUE	DESCRIPTION
Type	1	Decimal Number
	a	Alphabet Lower
	A	Alphabet Upper
	i	Roman Lowercase
	I	Roman Uppercase
	reversed	Reverse Order List
Start	1	Default is SET at 1

Learner's Hub
(B Praveen Kumar)

Un – Ordered List in HTML

Grouping of Elements in the form of <u>Disk</u> , <u>Circle</u> , <u>Square</u> are Considered as **Un-Ordered List**

<ul> Tag is Used as Un – Ordered List
    <li> Tag is also Child tag For Un – Ordered List

ATTRIBUTES	VALUE	DESCRIPTION
Type	Disc Circle Square	Disk Bullets Circle Bullets Square Bullets

Learner's Hub
(B Praveen Kumar)

Nested List in HTML

Definition List in HTML

A Nested List is a list within a List or Grouping of elements inside other List elements are Considered as **Nested List**.

Eg :

<ol> inside **<ol>**
or
<ol> inside **<ul>**
or
<ul> inside **<ul>**
or
<ul> inside **<ol>**

A definition list is similar to other lists but in a definition list, each list item contains two entries.. A **Term** and A **Description**.

Eg :

<dl> as Parent Definition List

<dt> as Definition Terms
<dd> as Definition Description

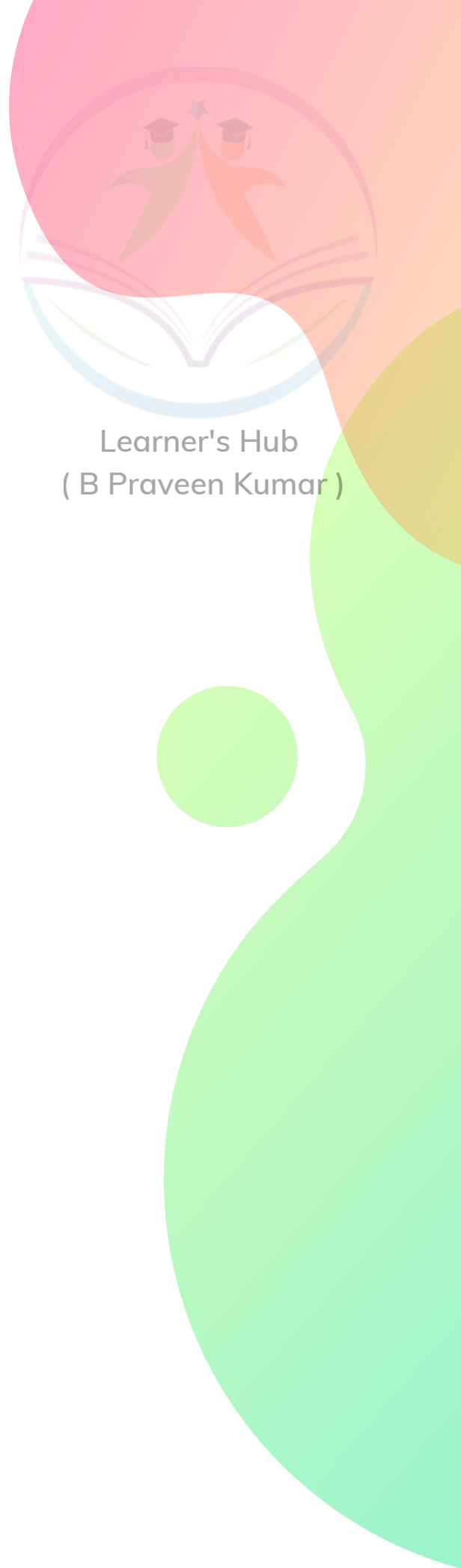**</dl>**

Learner's Hub
(B Praveen Kumar)

IMAGES in HTML

Images are Binary Formated Files which usually comes with
.jpg | .jpeg | .png | .gif as there Extensions

Images on HTML Pages can be Added 2 Ways..

Absolute Path : it is a path which can access image anywhere within the Local computer (c: or d:) it can also access External url's (http://) And can also access External Devices (Pen Drives , etc) in simple it can access images anywhere if Exist and is considered as Absolute path...

Relative Path : it is a Path which can access only its parent directory where html file saved... Cannot access out of its Parent Directory. And also will not work external url's and external Devices and is considered as relative path

To Add an Images on HTML Pages we might use **\<img\>** Tag

<u>Syntax</u> :

\<img src=' path of image ' /\>

<u>Attributes we may Pass to Images :</u>

Attribute Name	Attribute Description
Width = (num)	Controls Width of Image based on Argument Num We Pass
Height = (num)	Controls Height of Image based on Argument Num We Pass
title	Display Content when Mouse Hovers
border	Adds Border on top of the Image.
alt	Display Alternate Content only visible when image is not Found
hspace = (num)	Creates Horizontal Distance between images
vspace = (num)	Creates Vertical Distance between images

Learner's Hub
(B Praveen Kumar)

Entity in HTML

Entity are Special SYMBOL which are not available on the Keyboard.. And to represent such symbols we may Use Entity

Eg : 😃 Ⓒ ¥

HTML Supports 2 Types of Entity

∿ Named Entity

Describing Special Symbols using Names are Considered as **Named Entity**

Eg : _Ampersand + ENTITY NAME + Semi Colon_

© (represents Copyright Symbol)
€ (represents Euro Symbol)

∿ Numbered Enity

Describing Special Symbols using ASCII Number is Considered as **Numbered Entity**

Eg : _Ampersand + Hash + ASCII No + Semi Colon_

© (represents Copyright Symbol)

ANCHOR Tags in HTML

Click here

Anchor Tags are mainly used to redirect the links which can redirect **Internal or External Links**
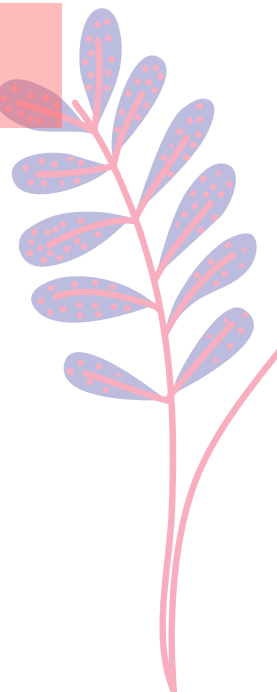
Using Anchor Tags we may redirect 4 Ways

    ➡️ External Links
    ➡️ Internal Links
    ➡️ Mail Links
    ➡️ Stay on the Same Page And redirect the links

Syntax :

<a href = ' path of link ' > Visibility to Front End ****

Attribute Name	Attribute Description
target = " _self | _parent "	Default – Opens Link in Same Browsers Screen
target = " _blank | _new "	Opens Links in new tab or new Window Screen
title	Display Content when Mouse Hovers

Videos in HTML

To Display **Video** on HTML page we might use **`<video>`** tag.

Syntax : **`<video src`** = ' path of video' **`></video>`** ↳ Note : mp4 as an ext

Attribute Name	Attribute Description
Wiidth = (num)	Controls Width of the Video
Height = (num)	Controls Height of the Video
controls	Adds Buttons on top of the Video Such as (Play , Pause , Zoom , etc)
muted	Volume will be SET to ZERO
loop	Plays Video by itself once video got Finished
autoplay	Plays Video by itself once page got Loaded
poster	(Path of image) Display image on top of the Video before Video got Started.

Audio in HTML

To Display **Audio** on HTML page we might use **<audio>** tag.

Syntax : **<audio src** = ' path of audio' **></audio>** ↳ Note : mp3 as an ext

Attribute Name	Attribute Description
controls	Adds Buttons on top of the Audio Such as (Play , Pause , Zoom , etc)
muted	Volume will be SET to ZERO
loop	Plays Audio by itself once audio got Finished
autoplay	Plays Audio by itself once page got Loaded

Tables in HTML

Table allows you to arrange data into rows and columns. They are commonly used to display tabular data like product listings, customer's details, financial reports, Accounts, Database Details or GMAIL inbox and so on.

HTML Table Tags

Table Tag	Table Description
\<table\>	Allows to Create Tables on HTML Page
\<tr\>	Use to Create Table Row
\<th\>	Use to Create Table Heading Title
\<td\>	Use to Create Table Column which Holds a Data Content
\<caption\>	Use to Create Table Caption
\<colgroup\>	Use for Creating one or more Group Column

<col>	Used with **<colgroup>** Element to Specify each Column
<thead>	Holds **Head** Section of Tables
<tbody>	Holds **Body** Section of Tables
<tfoot>	Holds **Foot** Section of Tables

Learner's Hub
(B Praveen Kumar)

Table Attributes	Attributes Description -- [Deprecated attributes]
align	Table Alignment [left* , center , right]
valign	Table Vertical Alignment [top , middle* , bottom]
border = (num)	Adds Border on top of the Table
width = (num)	Controls Width of Tables
bgcolor	Adds Background Color to the tables
background	(path of image) Display images on top of the table
Cellpadding = (num)	Create Distance between content & Inner Wall
Cellspacing = (num)	Controls Inner thickness of wall or border.

Again
Deprecated

colspan = (num)	Converts no of columns as Single column
rowspan = (num)	Converts no of Rows as Single row
frame	none | lhs | rhs | above | below | vsides | hsides
rules	none | cols | rows

Forms in HTML

HTML Forms are required, when you want to collect some data from the site visitor.
For example, during user registration you would like to collect information such as name, email address, credit card, etc.

Using Forms we may Create **SignUp Form, Login Form, Contact Form, Subscription Form** or Complex Forms Like **Tax Forms , Loan Form , Insurance Form** , etc..

Syntax :

        <form> Tag

HTML Form Tags

Form Tags	Form Description
<form>	Defines a form for user input.
<input>	Defines an input field data.
<button>	Defines a push button
<textarea>	Defines a text-area (a multi-line box).
<label>	Defines a label to the description.
<fieldset>	Defines a border to the input data.
<legend>	Defines a caption name write into fieldset.
<select>	Defines a drop-down select list box.
<option>	Defines an option value in the drop-down box.

Learner's Hub
(B Praveen Kumar)

<form> Tag in HTML

Attributes	Value	Form Description
action	"Sender addresss" "Page" "Database_page"	Allows to Send Form Data Values to Redirected Links
method	"GET" "POST"	By Default Form travels through **GET** * & to Secure data we may Pass **POST.**
enctype	multipart/form-data	it is necessary if your users are required to upload a file through the form.
target	_self * _blank	Allows Form data Value travel on same Screen. if passed _blank form data values will travel on new tab.

HTML <input> Tag

Attributes	Value	Form Description
<input type	"text"	Use of input text value
	"password"	Use of input password value
	"file"	Use for Upload Files
	"checkbox"	Allows to Create Checkbox
	"radio"	Allows to Create Radio Options
	"button"	Allows to Create Buttons insde the form
	"hidden"	Allows the Form to hold Hidden data
	"reset"	Allows to Reset the Form
	"submit"	Allows values to be submitted.
value	"Value to be stored"	Hold Value of input Elements

Learner's Hub
(B Praveen Kumar)

Attributes	Value	Form Description
name	"Name Holder"	Holds the Value.
size	"Number"	Defines Char Size inside input box
maxlength	"Number"	Defines Max Char Size inside input box
checked		Makes Checkebox Checked.
alt	"text"	Defines Elements to identify
id	"Should be Unique"	Allow to Target Single elements to Validate or apply conditions using JavaScript

Learner's Hub
(B Praveen Kumar)

Attributes	Value	Form Description
align	"left" Default * "center" "right"	Defines Allignment
src	"url"	Defines Image on submit buttons
label	"id of the Element"	Creates relation between content & its input element.

Learner's Hub
(B Praveen Kumar)

HTML \<textarea\> Tag.

Attributes	Value	Form Description
cols	"specify_number"	Define the Specify number of character visible in one line of text area.
rows	"specify_number"	Define the Specify number of line visible in text area.
name	"unique_name"	Specify unique name for the input element.

Learner's Hub
(B Praveen Kumar)

HTML \<selection> Tag.

Attributes	Value	Form Description
multiple	"space"	Define the Multiple select of option list.
size	"specify_number"	Define the Specify number of item list size.
name	"specify_name"	Define the Specify name to select list description.

New Input Types in HTML5

HTML5 introduces several new <input> types like **email, date, time, color, range**, and so on. to improve the user experience and to make the forms more interactive. However, if a browser failed to recognize these new input types, it will treat them like a normal text box.

Learner's Hub
(B Praveen Kumar)

In this section we're going to take a brief look at each of the following new input types:

→ color   → date   → datetime-local

→ email   → month   → number

→ range   → search   → tel

→ time   → url   → week

⚠ **Warning:** The input `type="datetime-local"` is not supported by Firefox, Safari, and Internet Explorer browsers. Currently supported by Chrome, Edge, and Opera browsers.

HTML5 New Tags

This section contains a complete list of standard tags belonging to the latest HTML5 specifications

HTML 5 Tags	Description
<article>	Defines an article.
<aside>	Defines some content loosely related to the page content.
<details>	Represents a widget from which the user can obtain additional information
<hggroup>	Defines a group of headings.
<summary>	Defines a summary for the <details> element.

<datalist> — Represents a set of pre-defined options for an <input> element.

<keygen> — Represents a control for generating a public-private key pair.

<meter> — Represents a scalar measurement within a known range.

<bdi> — Represents text that is isolated from its surrounding for the purposes of bidirectional text formatting.

<bdi> — Represents a control for generating a public-private key pair.

<mark> — Represents text highlighted for reference purposes.

<output> — Represents the result of a calculation.

<progress> — Represents the completion progress of a task.

<rp> — Provides fall-back parenthesis for browsers that that don't support ruby annotations.

<rt> — Represents a control for generating a public-private key pair.

<ruby> — Represents a ruby annotation.

<wbr> — Represents a line break opportunity.

<audio> — Embeds a sound, or an audio stream in an HTML document.

<canvas> — Defines a region in the document, which can be used to draw graphics on the fly via scripting (usually JavaScript).

<embed> — Embeds external application, typically multimedia content like audio or video into an HTML document.

<figcaption> — Defines a caption or legend for a figure.

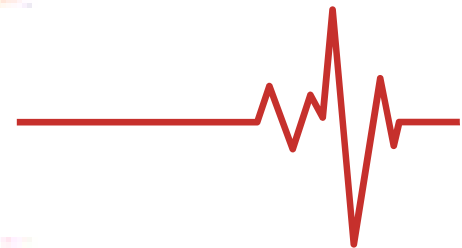**<figure>** — Represents a figure illustrated as part of the document.

<source> — Defines alternative media resources for the media elements like <audio> or <video>.
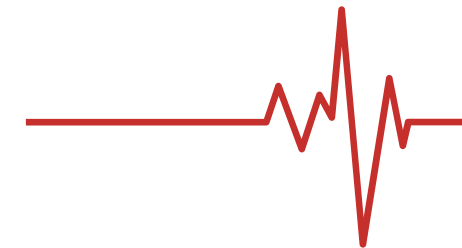
<time> — Represents a time and/or date.

<video> — Embeds video content in an HTML document.

HTML5 Layout Tags

<header> — Represents the header of a document or a section.

<footer> — Represents the footer of a document or a section.

<aside> — Defines some content loosely related to the page content.

<nav> — Defines a section of navigation links.

<section> — Defines a section of a document, such as header, footer etc.

<article> — Defines an article.

Learner's Hub
(B Praveen Kumar)

Canvas in HTML5

The HTML5 canvas element can be used to draw graphics on the webpage via **JavaScript.** The canvas was originally introduced by Apple for the Mac OS dashboard widgets and to power graphics in the Safari web browser. Later it was adopted by the Firefox, Google Chrome and Opera. Now the canvas is a part of the new HTML5 specification for next generation web technologies.

By default the **<canvas>** element has 300px of width and **150px** of height without any border and content. However, custom width and height can be defined using the CSS height and width property whereas the border can be applied using the CSS border property.

Understanding Canvas Coordinates

The canvas is a two-dimensional rectangular area. The coordinates of the top-left corner of the canvas are (0, 0) which is known as origin, and the coordinates of the bottom-right corner are (canvas width, canvas height). Here's a simple demonstration of canvas default coordinate system.

(0,0)

X

Y

(0,0)

The coordinates of the top-left corner of the canvas are (0, 0)

The coordinates of the bottom-right corner are (canvas width, canvas height)

SVG in HTML5

The Scalable Vector Graphics (SVG) is an XML-based image format that is used to define two-dimensional vector based graphics for the web. Unlike image (e.g. .jpg, .gif, .png, etc.), a vector image can be scaled up or down to any extent without losing the image quality.

There are several other advantages of using SVG over other image formats like JPEG, GIF, PNG, etc.

→ SVG images can be searched, indexed, scripted, and compressed.

→ SVG images can be created and modified using JavaScript in real time.

→ SVG images can be printed with high quality at any resolution.

→ SVG content can be animated using the built-in animation elements.

→ SVG images can contain hyperlinks to other documents.

Differences between SVG and Canvas

The HTML5 introduced the two new graphical elements <canvas> and <svg> for creating rich graphics on the web, but they are fundamentally different.

Learner's Hub
(B Praveen Kumar)

SVG	CANVAS
→ Vector based (composed of shapes)	Raster based (composed of pixel)
→ Multiple graphical elements, which become the part of the page's DOM tree	Single element similar to <img> in behavior. Canvas diagram can be saved to PNG or JPG format
→ Modified through script and CSS	Modified through script only
→ Good text rendering capabilities	Poor text rendering capabilities

SVG	CANVAS
→ Give better performance with smaller number of objects or larger surface, or both	Give better performance with larger number of objects or smaller surface, or both
→ Better scalability. Can be printed with high quality at any resolution. Pixelation does not occur	Poor scalability. Not suitable for printing on higher resolution. Pixelation may occur

HTML 5 WebStorage

The HTML5's web storage feature lets you store some information locally on the user's computer, similar to cookies, but it is faster and much better than cookies. However, web storage is no more secure than cookies.

The information stored in the web storage isn't sent to the web server as opposed to the cookies where data sent to the server with every request. Also, where cookies let you store a small amount of data (nearly 4KB), the web storage allows you to store up to 5MB of data.

There are two types of web storage, which differ in scope and lifetime:

Local storage — The local storage uses the localStorage object to store data for your entire website on a permanent basis. That means the stored local data will be available on the next day, the next week, or the next year unless you remove it.

Session storage — The session storage uses the sessionStorage object to store data on a temporary basis, for a single browser window or tab. The data disappears when session ends i.e. when the user closes that browser window or tab.

HTML 5 The localStorage Object

the localStorage object stores the data with no expiration date. Each piece of data is stored in a key/value pair. The key identifies the name of the information (like 'first_name'), and the value is the value associated with that key (say '**Tina**'). Here's an example:

JavaScript code has the following meaning:

- **localStorage.setItem(key, value)** stores the value associated with a key.

- **localStorage.getItem(key)** retrieves the value associated with the key.

You can also remove a particular item from the storage if it exists, by passing the key name to the **removeItem()** method, like **localStorage.removeItem("first_name")**

However, if you want to remove the complete storage use the clear() method, like **localStorage.clear()**. The **clear()** method takes no arguments, and simply clears all key/value pairs from localStorage at once, so think carefully before you using it.
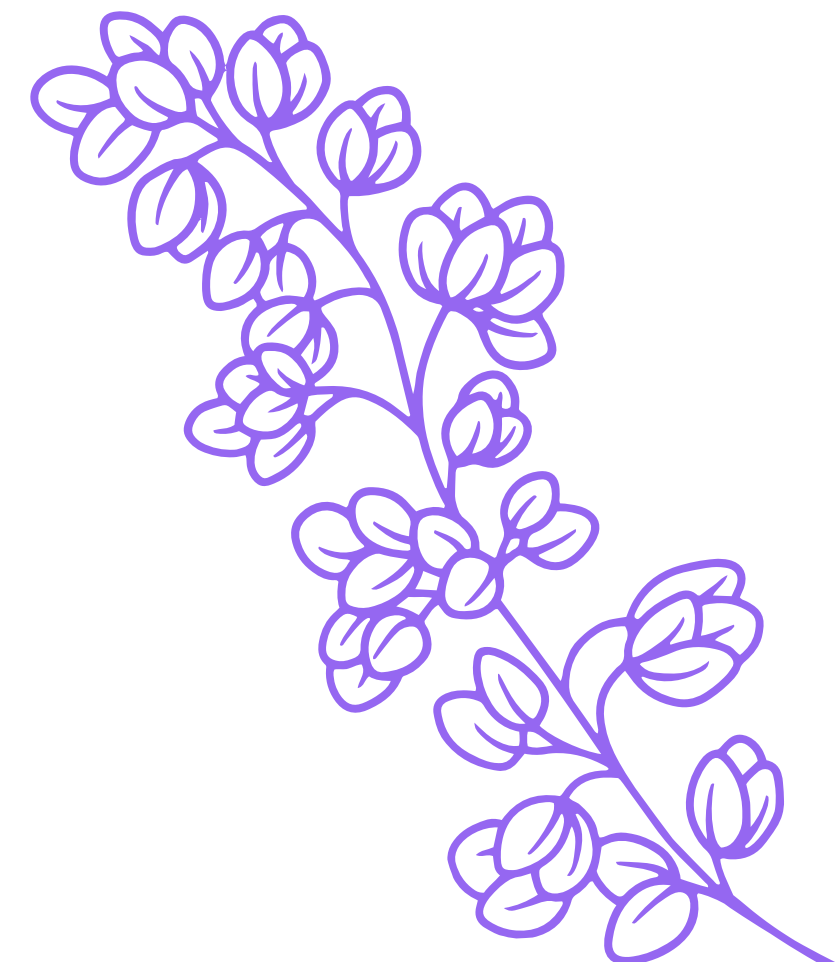
HTML 5 The sessionStorage Object

The sessionStorage object work in the same way as *localStorage*, except that it stores the data only for one session i.e. the data remains until the user closes that window or tab.

Note: The web storage data (both `localStorage` and `sessionStorage`) will not be available between different browsers, for example the data stored in Firefox browser will not available in Google Chrome, Safari, Internet Explorer or other browsers.

Let's try out the following example to understand how it basically works:

HTML 5 Application Cache

Typically most web-based applications will work only if you're online. But HTML5 introduces an application cache mechanism that allows the browser to automatically save the HTML file and all the other resources that needs to display it properly on the local machine, so that the browser can still access the web page and its resources without an internet connection.
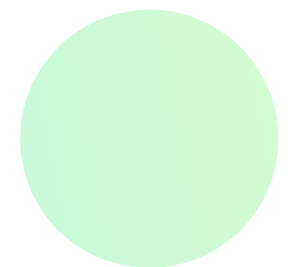
Here are some advantages of using the HTML5 application cache feature:

→ **Offline browsing :**

> Users can use the application even when they're offline or there are unexpected disruptions in the network connection.

→ **Improve performance :**

> Cached resources load directly from the user's machine rather than the remote server hence web pages load faster and performing better.

→ **Reduce HTTP request and server load :**

The browser will only have to download the updated/changed resources from the remote server that minimize the HTTP requests and saves precious bandwidth as well as reduce the load on the web server.

HTML 5 Geolocation

The HTML5 geolocation feature lets you find out the geographic coordinates (latitude and longitude numbers) of the current location of your website's visitor.

This feature is helpful for providing better browsing experience to the site visitor. For example, you can return the search results that are physically close to the user's location.

Finding a Visitor's Coordinates

Getting the position information of the site visitor using the HTML5 geolocation API is fairly simple. It utilizes the **three methods** that are packed into the **navigator.geolocation** object

→ **getCurrentPosition()**
   it returns a coordinates object to the function specified in the in the parameter (showPosition)

→ **showPosition()**
   it returns outputs of the Latitude and Longitude value

→ **watchPosition()**
   Returns the current position of the user and continues to return updated position as the user moves (like the GPS in a car).

→ **clearWatch()**
   Stops the watchPosition() method.

Learner's Hub
(B Praveen Kumar)

HTML5 Server-Sent Events (SSE)

HTML5 server-sent event is a new way for the web pages to communicating with the web server. It is also possible with the XMLHttpRequest object that lets your JavaScript code make a request to the web server, but it's a one-for-one exchange — that means once the web server provides its response, the communication is over. XMLHttpRequest object is the core of all Ajax operations.

However, there are some situations where web pages require a longer-term connection to the web server. A typical example is stock quotes on finance websites where price updated automatically. Another example is a news ticker running on various media websites.

Possible Applications

→ A real-time chart of streaming stock prices

→ Real-time news coverage of an important event (posting links, tweets, and images)

→ A monitor for server statistics like uptime, health, and running processes

Overview of the API

→ **new EventSource(url)**

This creates our EventSource object, which immediately starts listening for events on the given URL.

→ **onopen, onmessage**

two events that we can listen for on the new EventSource object. By default, the message event will fire when new messages are received, unless the server explicitly sets the event type.

→ **addEventListener**

not only can we listen for the default message event, but we can also listen for custom messages using the addEventListener on the EventSource object, just as if we were listening for a click event.

→ **event.data**

as with most messaging APIs, the contents of the message reside in the data property of the event object. This is a string, so if we want to pass around an object, we need to encode and decode it with JSON.

→ **close**

closes the connection from the client side.

HTML5 Drag and Drop

The HTML5 drag and drop feature allows the user to drag and drop an element to another location. The drop location may be a different application. While dragging an element a translucent representation of the element is follow the mouse pointer.

Tip: You can make an element draggable by setting the draggable attribute to true, like `draggable="true"`. However, in most web browsers, text selections, images, and anchor elements with an `href` attribute are draggable by default.

Drag and Drop Events

A number of events are fired during the various stages of the drag and drop operation. But mouse events such as mousemove are not fired during a drag operation.

The following table provides you a brief overview of all the drag and drop events.

Event	Description
→ ondragstart	Fires when the user starts dragging an element.
→ ondragenter	Fires when a draggable element is first moved into a drop listener.
→ ondragover	Fires when the user drags an element over a drop listener.
→ ondreagleave	Fires when the user drags an element out of drop listener.
→ ondrag	Fires when the user drags an element anywhere; fires constantly but can give X and Y coordinates of the mouse cursor.
→ ondrop	Fires when the user drops an element into a drop listener successfully.
→ ondragend	Fires when the drag action is complete, whether it was successful or not. This event is not fired when dragging a file to the browser from the desktop.