HTML

Q1. What is Hyper Text?

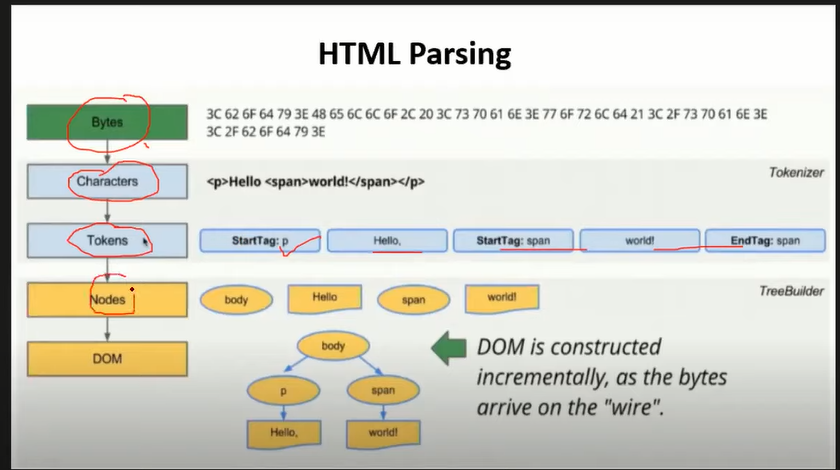
* The term “hyper” means beyond
* It specifies that text contains information beyond what you see.

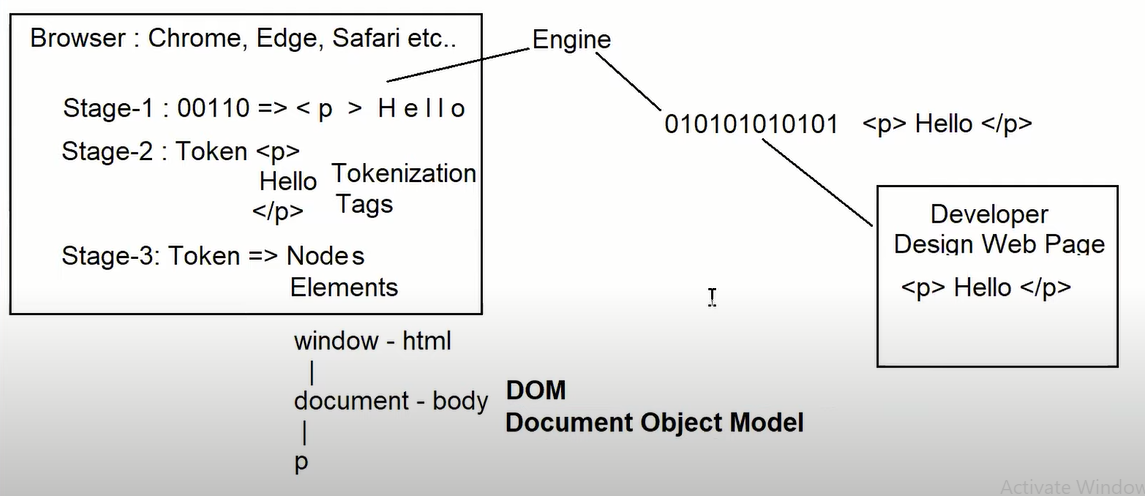
Q2. What is a Markup Language?

* The term markup is derived from “Marking Up”, which means preparing content for presentation.
* It is a language used for presentation.
* It is used to present on browser.

Evolution of HTML

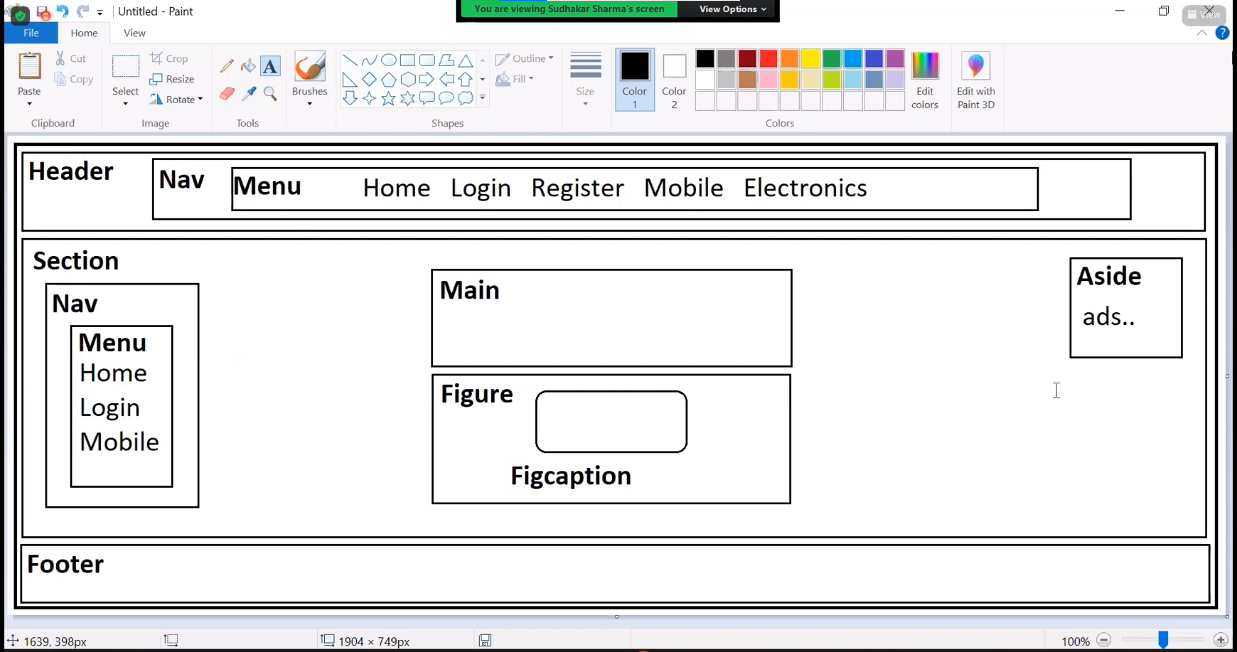
* GML [Generic Markup Language) is the first markup language used on Internet. [Developed by CERN Labs]
* "Council for European Research and Neuclear"
* SGML (Standard Generic Markup Language]
* 1990 - Tim Berners Lee introduced "HTML".
* IETF [Internet Engineering Task Force) developed HTML upto 3.2 version.
* 2004 Whatwg and W3C started evolution of HTML as version 4.0
* Web Hyper Text Application Technology Work Group [WHATWG] World Wide Web Consortium [ W3C]
* The latest version of HTML is "HTML 5".





**Body Semantic Elements**

* Upto HTML 4 body section is designed by using Tables.
* Tables lead to a situation called "Kiss-of-Death".
* Tables are not SEO friendly. - HTML 5 Introduced new elements into Body section to design a layout that can be SEO friendly.
* The new elements of HTML 5 Body section are:

**1. aside**

**2. article**

**3. dialog**

**4. figure**

**5. figcaption**

**6. header**

**10. nav**

**11.menu**

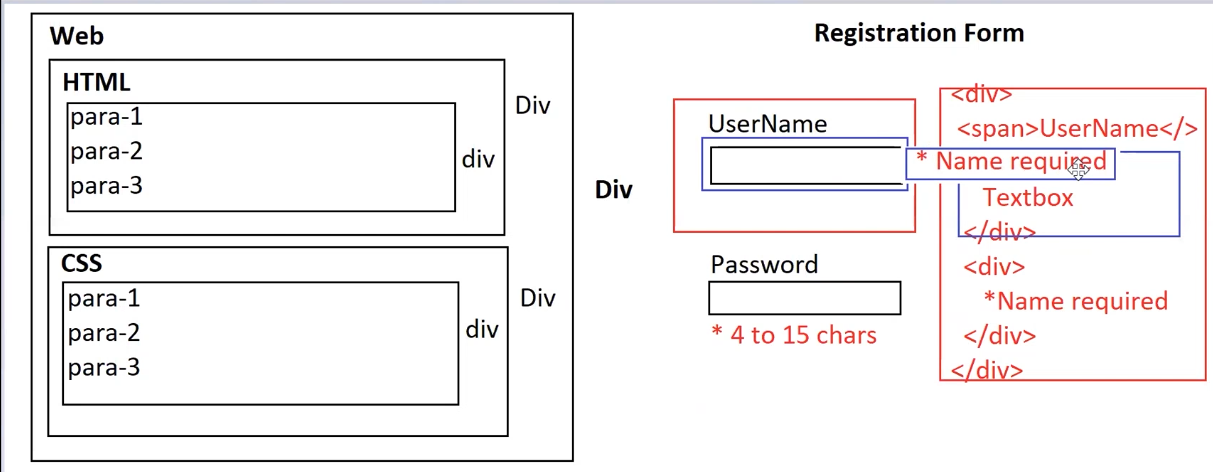
**12.div**

**13.span**

* + All are containers

**<aside>**

**...your content**

**<laside>**

* + Div defines blocked content.
  + Span defines inline content.

1. Add <style> container in

<head> section

<head>

<style>

</style>

</head>

1. You can add styles

into styles container

selector {

attribute : value

}

Note: Select defines where you want to apply effects, to which element.

* + **Style Attributes** 
    - Background-color : set background color
    - color : set text color
    - text-align : set alignment left, center, right, justify
    - border : set border [size, style, color]
    - border : 2px solid dotted dashed red margin-left margin-right
    - margin-left margin-right margin-top margin-bottom
    - margin : all directions
    - padding-left padding-right padding-top padding-bottom
    - padding : all directions
* **Distribute Contents into multiple Columns**
* display:grid
* grid-template-columns: how many columns and their size. Max 12 columns
* Note: Column width in pixels or fractions
* pixels : max 1200px fixed size
* fr: max 12 adjust according to browser
* Columns can be defined by using:
  + display: grid
  + columns
* Display:Grid
  + Every column is individual.
  + Contents of one column will not span to next column.
* Columns :
  + It is used for continous columns.
  + The content will span to next column when it fits the previous.
* Syntax:

#Container {

columns : 3;

}

#Container {

display:grid;

grid-template-columns: 2fr 10fr

}

* **Responsive Navigation:**
* Navigation bar must change its orientation according to device screen.
* You can define it by using CSS @media queries
* Syntax:

@media screen and (orientation:landscape portrait)

**Adding Icons to Page**

* Download and Install "bootstrap icons"
* Open Terminal Switch to Command Prompt
* Type the following commmand

> npm install bootstrap-icons

* All icons are kept in a ".css" file.
* You have to link the CSS file to webpage.
* Every Icon will have a class name, you have to apply by using "Class" attribute.

<div class="bi bi-house">

GET Help about icons from [www.getbootstrap.com](http://www.getbootstrap.com)

* Example

1. Open Terminal

> npm install bootstrap-icons --save

2. Go to HTML file and link stylesheet

<link rel="stylesheet"

href="../node\_modules/bootstrapicons/font/bootstrap-icons.css">

3. You can apply icon to any container

<span class="bi bi-alarm">

<div class="bi bi-alarm">

* Mouse Over Effect:
* You can apply by using ":hover"
* Syntax:

div { } - normal

div :hover { } - on mouse over

**Body Section Literals**

* **Line Breaks** <br>
  + FAQ: What is difference between <br> and <br />?
  + No difference. HTML will not provide <br />. It is just developers technique of writing void element.

<!-- pre formatted text -->

<p>Simple C program</p>

<pre>

#inclue <stdio.h>

main(){

printf("Hello World!");

}

</pre>

<!-- variables -->

var <var>x</var> = 10;<br>

var <var>y</var> = 20;<br>

var <var>z</var> = <var>x</var> + <var>y</var>;<br>

<!-- code snippet -->

<code> Page Not Found-http://127.0.0.1:5500/public/5\_body\_literals.html</code>

<!-- Titles for content -->

<code><span title="Page not available">404</span>-http://127.0.0.1:5500/public/5\_body\_literals.html</code>

* **Blank Spaces** &nbsp

[non-breakable space]

* **Pre Formatted Text**

<pre>

</pre>

* **Variables** <var>
* **Code Snippet**

<code>

</code>

* **Titles for content** 
  + Title is a screen tip display

for text, image or any graphic.

<span title=""> Text </span>

* **Headings** 
  + <h1> …….</h1> first level
  + <h6>……..</h6> last level
  + Headings can use align attribute for left, center, right or justify.

<h1 align="center"> </h1>

* + Don't use headings to highlight any word or sentence in paragraph.
  + Don't use too many headings in page. It may lead to SPAM.

FAQ: Why to use heading element? Without heading we can apply similar effect.

Ans : To make the page topics SEO friendly.

FAQ: Can we change the font size and color or effects for heading?

Ans : Yes, With styles.

* **Paragraphs and Blockqotes**

<p> Paragraph

<blockqouote> Blockqoute

* You can align paragraph by using left, center, right, justify.
* **Data List in HTML** 
  + It is a collection of terms and definitions.
  + Terms are defined by using <dt>
  + Definition defined by using <dd>
  + Terms and definitions are defined in <dl>
  + Syntax

<dl>

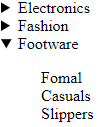
<dt>Term</dt>

<dd>Definition</dd>

<dt>Term</dt>

<dd>Definition</dd>

</dl>

* **Details and Summary** 
  + Details are a collapsible container.
  + It is defined by using <details>
  + Details container can contain any kind information within the context.
  + Details uses <summary> to display title for details your defined.
  + Syntax

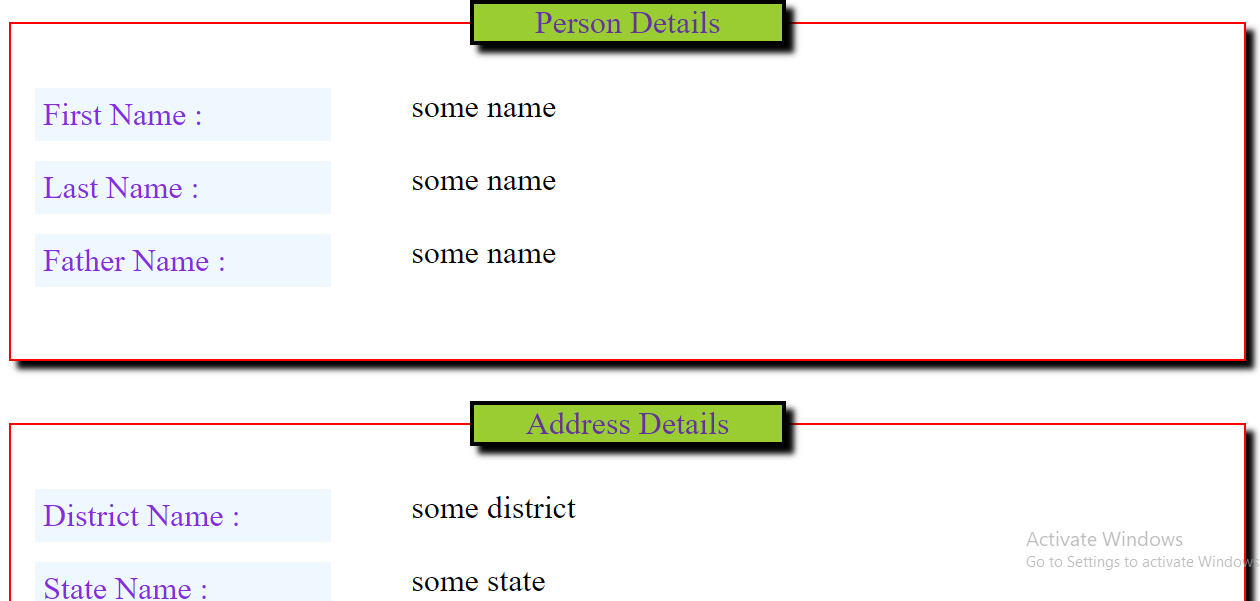
<details>

<summary> Title </summary>

...any content....

</details>

* You can use “open” attribute for details to keep the details open always.

<details open>

* **Fieldset and Legend:** 
  + Fieldset is a container with frame.
  + It is defined by using <fieldset>
  + Every fieldset can have a <legend>
  + Legend summaries the topic in fieldset.
  + Syntax:

<fieldset>

<legend>Title</legend>

....your content...

</fieldset>

* CSS Shadow:
  + box-shadow : It is used for any container
  + text-shadow : It is only for text.
  + Syntax: box-shadow: hPixel, vPixel, blurPixel, color;
* **Ordered List** 
  + It is used for auto numbering.
  + It is configured by using <ol>
  + It comprises of collection of items defined with <li>.
  + Syntax:

<ol>

<li> Item-1 </li>

<li> Item-2 </li>

</ol>

* + The numbering type can be defined by using "type" attribute

<ol type="1, A, a, 1, 1">

* + You can define the level number to start with.

<ol start="5">

<ol type="a" start="5">

* + You can also reverse numbering

<ol reversed>

* + - **Nested Order List :** Configuring a list within another list.

Note : You have to configure a <ol> within <li> not below

Good

<ol>

<li> Item

<ol>

<li>child</li>

</ol>

</li>

</ol>

Not Good

<ol>

<li> Item</li>

<ol>

<li>child</li>

</ol>

</ol>

**FAQ : How to design List items side by side?**

Ans : by using **display : flex**

<ol>

<li> Java is a \_\_\_\_\_

<ol type=”a” id=”childList”>

<li>Prog. Lang.</li>

<li>Techno.</li>

<li>compiler</li>

</ol>

</li>

</ol>

**<style>**

**#childList{**

**display : flex;**

**margin-bottom : 20px;**

**}**

**li{**

**margin-right : 40px**

**}**

**</style>**

**FAQ : How to define list items in multiple columns?**

Ans : **display : grid (or) columns**

ol{

columns : 2;

}

#childList{

display : grid;

grid-template -columns : 6fr 6fr;

}

(OR)

**FAQ: How to design a scrollable List?**

Ans : By using **“overflow : auto”**

ol{

border : 2px solid black;

width : 150px;

height : 150px;

overflow : auto;

}

* **Unorderd List**
  + It is a bulleted list.
  + It comprises a symbol instead of number.
  + It is designed by using <ul> for unordered list and <li> for list item.
  + Syntax:

**<ul>**

**<li>Item-1</li>**

**</ul>**

* + You can change bullet symbol by using "type" attribute with values

disc, circle, square

**<ul type = “disc|circle|square”></ul>**

* + - You can have nested unordered list.
    - You can use all effects similar to <ol>
    - You can define assorted nested list.

**FAQ : How to define custom bullet symbols?**

Ans:By using CSS property **“list-style-image”**

[It can be applied for ul and ol]

ol{

list-style-image : url(“image-path”);

}

**FAQ: How to remove bullet or numbered symbol for list.**

Ans : By using CSS Property **“list-style:none”.**

**FAQ: Can we define different symbols for each list item individually?**

Ans : No, You have to use external icons library.[bootstrap-icons]

**Text Formatting in HTML**

* **Font** 
  + <font> element is used to change the character
    - Face
    - Size
    - color
  + Font face is the name of font family for character.
  + Syntax:

<font face="Brush Script MT"> Your Text </font>

**FAQ: What are web safe fonts?**

Ans: Web safe fonts are the standard fonts available for all devices.

The standard web safe fonts are:

a) monospace

b) sans-serif

c) serif

* Font Size can be defined in 7 levels
* From 1 to 7 size increases.

<font size = “4”>Your Text </font>

* Font color can be name or hexa code.

<font color = “red”>Your Text</font>

* **Font Styles** 
  + <b> Bold
  + <strong> Strong
  + <i> Italic
  + <em> Emphasized
  + Syntax:
    - <b> Text </b>
    - <strong> Text </strong>
    - <i> Text </i>
    - <em> Text </em>

**FAQ: What is difference between <b> bold and <strong> ?**

<b> is used in design mode to highlight

<strong> is used in review mode.

<i> is for design and <em> for review.

* **Font Effects**
  + <u> Underline
  + <ins> Inserted-Underline[Review]
  + <strike> Strikeout
  + <del> Deleted-Strikeout[Review]
  + <sup> Super script
  + <sub> sub script

**Images In HTML**

* Images are of various types.
* Web supports only few types.
* MIME is "Multipurpose Internet Mail Extensions".

|  |  |  |
| --- | --- | --- |
| **Image Type** | **MIME** | **Extension** |
| APNG | Image/apng  Animated Portable Network Graphics | .apng |
| PNG | image/png  Portable Network Graphics | .png |
| GIF | image/gif  Graphic Interchange Format | .gif |
| JPEG | image/jpeg  Joint Photographic Expert Group | .jpg, .jpeg, .jfif |
| TIFF | image/tiff  Tagged Image File Format | .tiff |
| ICO | image/icon  Microsoft Icon | . ico |
| SVG | image/svg  Scalar Vector Graphics | .svg |
| Webp | image/webp  Web Picture Format | .webp |

* **PNG | APNG** 
  + High Definition
  + High Resolution
  + More space in memory
  + Use PNG or APNG when you are providing images for download.
  + Not good for on screen presentation [more memory]
* **JPG | JPEG** 
  + High Definition
  + High Resolution
  + Compressed Image
  + Less space in memory
  + Good for on screen presentation
  + Not good for zoom and download.
* **SVG** 
  + It is XML format image
  + Vector Graphics
  + Not pixel based
  + You zoom in the picture, it quality will increase.
  + Good for architectural diagrams
* **Webp** 
  + It is web picture format
  + It can embed picture along with content and locks the content.
  + You can't download
* **TIFF** 
  + Sliced image format
  + Scanned documents Activate Windows
* You can embed images into web page by using <img> element.
* It is a void element.
* <img> Attributes:
  + alt : It specifies the alternative text to display when image fails to load.
  + src : It refers to the path and name of image.
  + Syntax:

<img alt="some text" src="..path">

* + title : It refers to the screen tip to display when mouse is over image.
  + Syntax:

<img alt="some text" src="..path" title="message">

* + width: It specifies width of image in pixels or percentage.
  + height: It specifies height of image in pixels or percentage.

Note: Size in % will adjust according to the browser.

It often called as "Fluid-Image".

<img width="300">

<img width="50%">

* align : It aligns the image left or right. Align is used to wrap text around.

<img align="left | right">

* hspace : Horizontal space around image
* vspace : Vertical space around image

<img vspace="45" hspace="20">

* border: Sets border for image.

<img border="2">

* **Advanced Attributes:**
* crossorigin : It is used when image is accessed from remote server.
* crossorigin = "anonymous use credentials"
* decoding : It defines how image have to load along with other contents.
* decoding = "async | sync"
* sync : load image and block others
* async : load along with other contents
* importance : It defines the priority of image.
* importance = low, high, auto

**Card Design with Images and Styles**

* Flexible Design
  + Display: Flex
  + Flex-Wrap:wrap - It will not disturb the size of element.
* To Keep content center on page.
  + display:flex
  + justify-content: center; Horizontal alignment
  + align-items:center; Vertical alignment

**Links in HTML**

* Link is clickable text, picture or graphic that navigates the user to any specific location.
* Every website must have good navigation techniques. - You can configure navigation using Links in HTML page.
* Links in webpage are also referred as "Hyper Link".
* Links are classified into 2 types
  + Intra document links
  + Inter document links
* Intra document links allow navigation from one location to another within the same page.
* Inter document links allow navigation from one document to another.
* Links are created by using <a> anchor element.
  + **Intra Document Links:** 
    - Configures navigation from one location to another within same page.
    - You have to define a unique ID for target location.

<h2 id="home">

<img id="pic">

<ul id="list">

* User can reach to specific location by requesting the ID reference from URL
* http://127.0.0.1:5500/public/shopping.html#home
* You can configure <a> anchor element that can reach to id.
* <a href="#home"> Text Image Symbol </a>

**FAQ: How can we change color for Links?**

Ans: By using CSS classes

a:link { } all normal links

a:visited { } only for visited links

a:active {} only for active links

a{} all states normal, visited, active

**FAQ: How to remove underline for HyperLink?**

Ans: By using "text-decoration:none"

a{

text-decoration:none;

}

**FAQ: How to set underline on mouse over?**

Ans: By using "hover" class.

a:hover {

text-decoration:underline;

}

**FAQ: If content is not scrollable then how to highlight the active topic?**

Ans: By using **CSS “target” class.**

.topic{

Grey, white

}

.topic:target {

Black, white

}

* **Inter document links**
  + - Navigation from one page to another.
    - It can also navigate to any specific URL.
    - It can also navigate to any specific file.
    - It can also open any specific application.

Note:

The target file will open in browser or download it depends on plug-ins.

If plug-in is not available or supported then file can't open, it will always download.

**FAQ: How to design a link that always download even when plugin is available?**

Ans: By using **"download" attribute for <a>**

Syntax:

<a href="../public/images/shoe.jpg" download="nike"> Nike </a>

**FAQ: How do you link to any application?**

Ans: By using **"href" with following attributes**

mailto :email app

tel : phone dial app

skype : skype app

javascript : for any another app or function

Syntax:

<a href=”<mailto:email@address>”></a>

**FAQ: How to open link in a new tab?**

Ans: By **using "target" attribute set to "\_blank".**

target=\_blank

**FAQ: How to open link in a new window?**

Ans: By **using "javascript" window.open() method.**

Syntax:

<a href="javascript:window.open('path', 'title','width-height')">File Name</a>

**FAQ: How to open in same window along with existing content?**

Ans: By **using <iframe> of HTML, that used target attribute with frame name.**

**FAQ: What is iframe?**

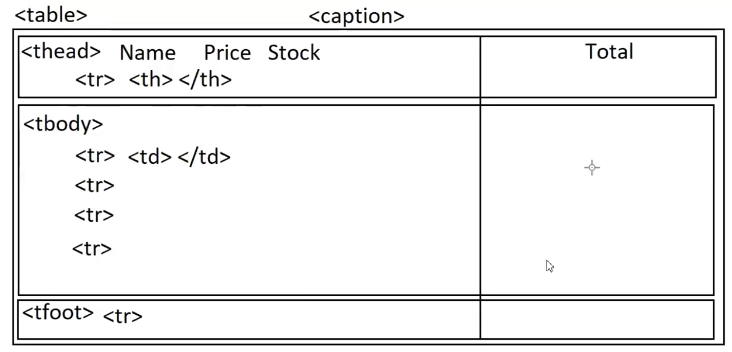
Ans: Iframe is used to embed any external content into page, which can be image, video, animation, document, etc..

Syntax:

<iframe src="filepath"> </iframe>

* Summar
* Open in same window
* Open in new tab
* Open in different window
* Open in same window along with other content[iframe]

**Tables in HTML**

* Tables are used to store data in rows and columns. Table Elements are
  + <table>
  + <caption>
  + <thead>
  + <tbody>
  + <tfoot>
  + <colgroup>
  + <th>
  + <tr>
  + <td>
* **Table Attributes** 
  + Border, Frame and Rules
    - Frame :
      * It is used for table, you can apply values
      * box | rhs | Ihs | above | below | void
      * Syntax:

<table frame="box">

* + - Rules :
      * It is used for rows, columns and groups
      * rows cols | groups | none | all
      * Syntax:

<table frame="box" rules="groups">

* + - Border :
      * It is used for cell. It can have values 0 or 1.
      * 0 = false [no border]
      * 1 = true [ border]

**Note**: Border can't display if rules are defined. It is mandatory to remove rules attribute from table.

**<table border="1" rules="none">// invalid**

**<table border="1"> //valid**

* Cellspacing and Cellpadding
* cellspacing : It sets space between cells.
* cellpadding : It sets space between border and content.
* Syntax:

<table cellspacing="10" cellpadding="5" border="1" frame="void">

* + bgcolor and background
* bgcolor :
* sets background color for table, group row or cell.

<table bgcolor="yellow">

<thead bgcolor="red">

<tr bgcolor="yellow">

<td bgcolor="red">

* background :
* sets background image for table, group row or cell.

<table background=”imagePath”>

* + align and valign :
* align : Horizontally left, center, right or justify
* valign : Vertically top, center, bottom.
* align => table, tr, group, cell
* valign => tr, group, cell
* Syntax:

<table align="center">

<tr align="center" valign="top">

* width and height
* width : It sets width for table and cell.
* height : : It sets height for table, group, row, cell
* Syntax:

<table width="300">

<th width="100">

<td width="200">

<tr height="100">

<td height="100">

* colspan and rowspan
* colspan : It is used to merge multiple columns into single cell.
* rowspan : It is used to merge multiple rows into single cell.
* **Nested Tables**
* You can define a table inside another table.

<tr>

<td>

<table/>

<table/>

</td>

</tr>

**Forms in HTML**

* Form is a container that provides various controls to interact with application and data.
* Form provides an UI for interactions like view, edit, update, insert and delete
* Form comprises of various types of controls like
  + buttons
  + textbox
  + checkbox etc…
* **Form Container :**
  + Form container is designed by using

<form> </form>

* Form comprises of various attributes
  + name
  + id
  + class

<form id="frmRegister" name="Register Form" class="formregister"> </form>

* method :
* It defines the actions performed by form.
* HTML form can be designed for 2 methods
* GET
* POST
* GET is used to fetch information from server.
* POST is used to submit information to server.

Note: The default method for form is GET.

**FAQ: Can we submit information on GET?**

Ans: Yes. But not recommended.

* GET vs POST
* **GET** submits data as querystring.
* Data will be appended into URL
* Anyone can view your data.
* It is not safe
* Easy to hack
* It is stored in browser logs.
* It can be bookmarked
* You can’t submit complex data, like binary data.
* Limit for data 2048 chars.
* You can cache the data[buffer]
* You can view data faster, it saves round trips.
* **Post** Data will be submitted into form body
* it is not in URL
* safer that get
* hard to hack not stored in browser logs
* you can't bookmark
* no limit, complex data
* can't be cached
* action :
* It indicates target location where the data need to be submitted.
* Syntax :

<form method = “POST” action = “serverPage”>…..</form>

<form method = “POST” action = “register.php”>…..</form>

* novalidate :
  + It is used to by-pass the default HTML validations.
  + Syntax :

<form method = “POST” novalidate>….</form>

* **Form Elements**
* A form comprises of multiple elements, which are designed by using

<input>

<select>

<option>

<optgroup>etc…

* **TextBox**
* Textbox is used for string input.
* String is literal with group of chars, A-Z,0-9, Symbols.
* Textbox is configure by using

<input type="text">

* **Note:** <input> element allows to input any value. If you want to restrict to any specific data type like number, email, string etc, then you have to use the attribute "type".
* **Text Box attributes**
* id
* name
* class

**Note:** Every element in form must have a "name" defined. If name is not defined then form can't submit its value.

* value : the default value to display in textbox.
* readonly: It will not allow to change the value. But it submits the value.
* disabled : It will not allow to change and submit.
* Syntax:

<input type="text" name="Username" disabled value="john">

* size : It specifies the width of textbox. [20]
* Syntax:

<input type="text" name="otp" size="4">

* autofocus: It sets focus to the control automatically.
* Syntax:

<input type="text" name="opt" size="4" autofocus>

* minlength : It validates the minimum number of chars.
* maxlength : It validates the maximum number of chars. It will not allow exceeding chars.
* Syntax:

<input type="text" name="UserName" minlength="4" maxlength="15">

* required : It will not allow to submit empty. It is used to define mandatory fields.
* Syntax:

<input type="text" name="UserName" required>

* list : It uses a data list. data list is used as auto complete text.

**Note:** Data List is designed by using <datalist> Data list contains options designed by using <option>

* Syntax:

<input type="text" name="search" list="terms">

<datalist id="terms">

<option> Term 1 </option>

</datalist>

* pattern : It is used to restrict the format of value entered into textbox.
* Pattern uses a Regular Expression.

<input type="text" pattern="regExp">

* Regular expression is built by using meta chars and quantifiers.
* **Meta Characters**

|  |  |
| --- | --- |
| Meta Characters | Purpose |
| ? | Zero or One occurrence of character  Ex:  Color  Colour |
|  |  |

* **Check Box**
* Everything similar to radio button
* It allows checking and unchecking without Mutex.
* Syntax:

<input type="checkbox">

* **Attributes:**
* name
* value
* checked

**FAQ : How to display checkbox list?**

Ans : By using **CSS overflow**.

**FAQ: How to define effects based on checked propery?**

Ans : By using **CSS “:checked” class** you can control effect.

Syntax :

checkbox : checked{

….

}

* **DrodownList**
* Allows user to select any one option from a group of choices.
* Dropdown is designed by using

<select>

* Options in dropdown are designed by using

<option>

* You can group options by using

<optgroup>

* Syntax:

<select>

<optgroup label="CategoryName">

<option> </option>

<option> </option>

</optgroup>

</select>

* Every option comprises of 3 attributes
* value : value to submit
* selected : It makes the option selected
* disabled : It will not allow to select.
* Syntax:

<option value="" selected>

<option disabled>

* Every option is RC Data type, which means it will not allow any symbols, markup or images. Only plain text with special characters.
* Options can display multi lingual content as chars.
* **ListBox**
* Everything same like dropdown.
* It allows selecting one or multiple.
* You can change dropdown into listbox by using
* size
* multiple
* Syntax:

<select size="4" multiple>

<optgroup>

<option value="" selected></option>

</optgroup>

</select>

* **Meter**
* It is used for grade meter.
* It is defined by using <meter>
* Attributes:
* min
* max
* value
* low
* high
* Syntax:

<meter min = “” max = “” value = “” low = “” high = “”></meter>

* **Progress**
* It is used to display status of task performed in page.
* Like downloading, uploading, copying etc.
* Syntax :

<progress min = “1” max = “10” value = “50”></progress>

* **TextArea**
* It is used for displaying multiline text.
* It is RC data types
* Syntax :

<textarea>…..</textarea>

* **Buttons**
* Buttons are used to confirm user actions.
* It includes
* Record Actions - Insert, Update, Delete, View, Edit
* Record Navigations - Prev, Next, First, Last
* Miscellaneous Actions - Print, Save, Open, Close, Register, Login etc..
* Button are classified into 2 types
* Generic Buttons
* Non-Generic Buttons
* Generic buttons have pre-defined functionality
* HTML 4
* <input type="submit"> : It can submit form
* <input type="reset"> : It will reset form
* Non-Generic buttons are the buttons without any functionality.
* HTML 5
* <button type = “submit”></button>

**Install Bootstrap**

* npm install bootstrap --save
* Link to Page

<link rel="stylesheet" href="../node\_modules/bootstrap/dist/css/bootstrap.css">

* Set Container

<body class="container-fluid">

* Bootstrap Form Classes:
* .form-control : textbox, password, number, email, date url, textarea
* .form-select : dropdown
* .btn : buttons
* .btn-{color} : primary, success, danger, warning etc.. dark, light, info
* .input-group : It is used to display all inline.
* .input-group-text : It is used for prefix and suffix.
* .input-group-lg : large
* .input-group-sm : small
* Traditional Form Elements
* Container : <div>
* Label : <label>
* Elements : Textbox, CheckBox, Button, Radio

**Multimedia in HTML**

* Embed audio and video content in page.
* Embed animation to page.
  + Marquee : It is used for displaying scrolling and sliding content on page.

<marquee>

your content

</marquee>

* Attributes
* scrollamount : It controls the speed 1 to 100
* direction : It controls scrolling direction left, right, up, down.
* behavior : It sets sliding alternate.
* Syntax:

<marquee behavior="alternate">

<marquee direction="right">

* loop : It specifies the number of times to loop marquee.
* width : content width.
* height : content height.
* bgcolor : background color.

Note: You can use only direction or behavior for marquee. Both are not allowed.

Only one can be defined

* Video and Audio
* <video> It is used to embed video files [.mp4, .mkv, .mov, .flv, .ogg, .avi]
* <audio> It is used to embed audio files [.mp3, .wav, .midi ]
* Syntax:

<video src="" poster="" controls> </video>

<audio src="" poster="" controls> </audio