Target to learn: HTML, CSS, **Java script**, Bootstrap, Typescript, **Angular/ ReactJS**.

We use above technologies for creating UI (User Interface) pages.

UI – User Interface – Which ever user experiences that page is called as UI page / Browser page.





**Validation?** – Checking the user entered details (user name, password, email, phone number etc..) at server side or client side is called as Validation.



Server or Client always perform validation on user given input data.

Response is two types🡪 Valid response / Error response.

Browser**?** 🡪 HTML page rendering/ displaying location. It internally has engine to execute UI technologies.



Browser:

How many browsers are there in the market? 🡪 So many browsers are available in the market.

What are they? Tell me 4 browsers?

Different companies created different browsers. Based on **browsing speed, memory usage** we use different browser.



Which technologies can a browser understand? **HTML, CSS, Javascript**.

A browser can execute/run only **HTML, CSS, Javascript** files.



Then what about other UI technologies? How they run in browser? ->

For example, Angular or ReactJs how they run in browser? 🡪 First, we convert/compile/transpile the angular/reactjs into JavaScript code then we run in browser.





What is meant by Extension 🡪 It is File format.

How to create these files? 🡺 Left/Right Arrow mark.

**<anyName>.extension**

<anyName>.html, <anyName>.css, <anyName>.js

Extension? -> Which tells which type/format of the file we are going to create.

Eg: for TEXT file we use extension as **.txt**

for WORD file we use extension as **.docx**

for PDF file we use extension as **.pdf**

for JAVA file we use extension as **.java**

for IMAGES file we use extension as **.jpeg, .png, .gif….**

for HTML file we use extension as **.html / .htm**

for CSS file we use extension as **.css**

for Javascript file we use extension as **.js**

**etc…………………………..**

**Abbreviations:**

**HTML 🡪 Hyper Text Markup Language.**

**CSS 🡪 Cascading Style Sheet.**

**HTML**

<https://www.w3schools.com/>

Browser engine will run/execute content on html page.

Content means? 🡪 Code we written.

**In HTML page we can write HTML code, CSS code and JS code combinedly.**

How many ways we can write HTML tags/CSS code/JS code inside html page?



What is html tag/element? Why we write tag?

How to write html tag? A tag contains open tag and closing tag.

|  |
| --- |
| **<tagName>** --> opening tag  content  **</tagName>** --> closing tag |

Tags we always write in NESTED format. Writing tags inside another tags. **Tag inside another tag.**

**Tag is also called as ELEMENT.**

HTML stands for Hyper Text Markup Language.

HTML is the standard markup language for creating Web pages / Websites.

HTML describes the structure of a Web page.

HTML consists of a series of elements.- Number of tags.

HTML elements tell the browser how to display the content.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Page Title</title>  </head>  <body>  <h1>My First Heading</h1>  <p>My first paragraph.</p>  </body>  </html> |

What is the basic structure of html page?

Root tag / parent tag.

Child tags.

The **<!DOCTYPE html>** declaration defines that this document is an HTML5 document

To convert our html page to HTML 5 version html page.

The <html> element is the root element of an HTML page

The <head> element contains **meta information** about the HTML page 🡪 Data about data-Meta data. High level information.

The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)

**The <body> element defines the document's body,** and is a container for all the visible contents, such as headings,

paragraphs, images, hyperlinks, tables, lists, etc.

\*\*Whatever the content we have written inside <title> tag will be displayed inside **window tab location**.

\*\*Whatever the content we have written inside <body> tag will be displayed inside **document location**.

**Brower objects:**

window\*\* 🡪 Overall browser is called as window.

document\*\*\*\*\*\* 🡪 White screen inside browser is called as document.

location 🡪 URL bar location is called as location.

Etc…



The <h1> element defines a large heading



The <p> element defines a paragraph

<br> 🡪 Empty tag. Used for the breaking the line.

<h1> 🡪 It is a Heading tag, To increase the size of the text we use heading tag.

Heading tag range is from <h1>…to <h6>.

Editors: notepad, notepad++, **VSCode,** Atom editor, Brackets….

We use VSCode, why VSCODE?

VSCode has auto completion feature. Automatically it prompts the required tags and their details.

What is meant by case sensitive? Is html being case sensitive? Is java being case sensitive?

Case sensitive means- upper case letters and lower-case letter.

For example: html engine treats <html> tag and <HTML> tag as same. There is No error.

<a> 🡪 Anchor tag/element. Use for hyperlinks.

**Every html tag has attributes.**

Attributes are used to provide additional information about HTML elements.

<img> 🡪 used to display images on document / browser.

|  |
| --- |
| **<img src="instagram2.ico" width="50" height="50" alt="instagram image...."/>** |

Here <img**/>** is a **self-closing tag**. Means we are not writing the closing tag. We are closing the tag in open tag only.

View HTML Source Code. 🡪 we can see all websites source code. right click on browser 🡪 select “view page source”

**Inspect** (**Developer Tool**) is a browser feature 🡪 right click on browser 🡪 select “Inspect”

We can see source code in “**Elements**” tab.

We can use JavaScript “**Console**”

We can see html page “**sources**”, means we can see source page location.

We can see request “**network**” information.

HTML attributes provide additional information about HTML elements.

Attributes are always specified in **the start tag.**

**title** attribute is used for tooltip. It always gives extra information.

**\*\* Id** attribute is used to identify tag/element uniquely. We should not duplicate the id **value** with other tags id values.

<p id=”**empId**” title=”My Title” class=”myClass”> This is My First Paragraph</p>

<p id=”**empId**” title=”My Title” class=”myClass”> This is My Second Paragraph</p> 🡪 WRONG

**Question:** What is the difference between <title> tag and title attribute?

we always write HTML Attributes in **key=”value” format** OR **htmlAttrubuteName=”value”** format. Eg: id=”empId”.

we always write CSS Attributes in **key: value format** OR **cssAttrubuteName:value** format. Eg: “color:red”; .

: 🡪 is called as Column

; 🡪 is called as Semi-Column.

**URL**: (**U**niform **R**esource **L**ocator)

1. **Absolute URL.** – Complete URL path which is available in another server / host.

<https://static.xx.fbcdn.net/rsrc.php/yB/r/2sFJRNmJ5OP.ico>

1. **Relative URL** - Links to an image that is hosted within the website. Here, the URL does not include the domain name.

Eg: src=”/images/facebook.jpeg”

Domain name? – <https://www.facebook.com> website name is called as domain name.

Style attribute:

So far, we know id attribute, class attribute, title attribute along with this we have another important attribute called “style”. Style means it is CSS style. Cascading **Style** Sheet.

The HTML style attribute is used to **add styles to an element, such as** **color, font, size**, and more.

Syntax:

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

Eg:

<p **style="color:red; font-size: 50px;**"> I am red </p> 🡪 Inline style.

|  |
| --- |
| <p **style="color:red; font-size: 50px;**"> I am red </p>  <p **style="color:red; font-size: 50px;**"> I am red </p>  <p **style="color:red; font-size: 50px;**"> I am red </p>  <p **style="color:red; font-size: 50px;**"> I am red </p>  <p **style="color:red; font-size: 50px;**"> I am red </p> |

Styles:

1. **Inline style** -> problem, code had to be duplicated.
2. **Internal style** -> <style> tag inside html page.
3. **External style ->** we create .css file and Import in all required .html pages.
4. Styles with JavaScript.

In order to overcome problem with inline style, they introduced internal style.

In internal style we are writing **<style>** tag/ element inside HTML page.

Eg: **test.html**

|  |
| --- |
| <style>          p {              background-color: green;              font-size: 50px;              width: 550px;              font-family: Verdana, Geneva, Tahoma, sans-serif;              height: 150px;          }      </style> |

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>      <title>My Page</title>  </head>  <style>  **p { // here P is called as “selector” 🡺 Find and Apply. Means find the selector and apply the CSS code.**      background-color: green;      font-size: 50px;      width: 550px;      font-family: Verdana, Geneva, Tahoma, sans-serif;      height: 150px;  }  </style>  <body>      <p>AAAAAAAAAAAAAAA</p>      <p>AAAAAAAAAAAAAAA</p>      <p>AAAAAAAAAAAAAAA</p>      <p >AAAAAAAAAAAAAAA</p>      <p>AAAAAAAAAAAAAAA</p>      <p>AAAAAAAAAAAAAAA</p>  </body>  </html> |

**Purpose of CSS “selector” is to find and apply the styles we defined.**

If we write inline style and internal style, which one takes priority? 🡪 Inline style only takes priority.

External style:

1. Custom .css/.js file. 🡪 code written by us.
2. Predefined .css/.js file. 🡪 code written by someone else and we are using it by importing in our html file.

Even though we write internal style still code is getting duplicated. How?

If we create multiple HTML pages with same CSS code, then css code getting duplicated.

To overcome this problem, they have given external style concept.

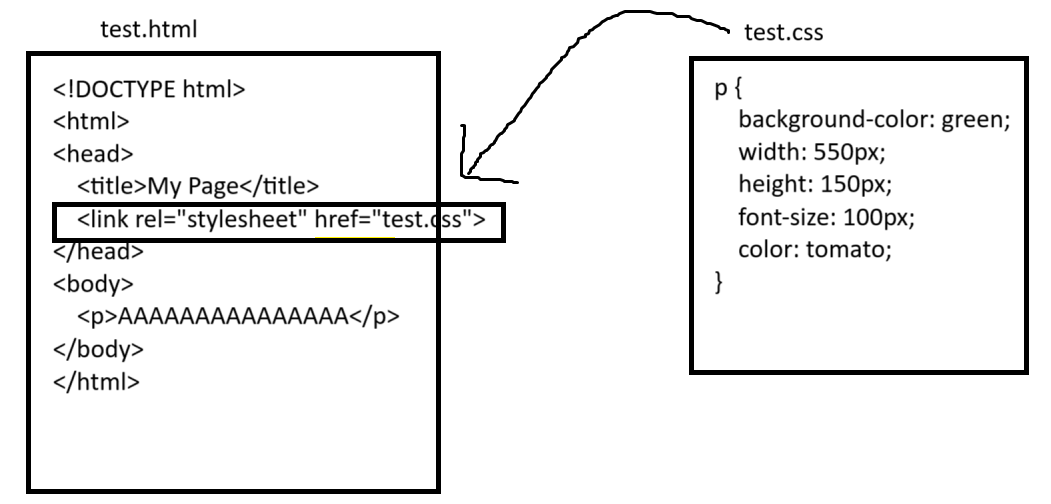
We write separate .ccs file and we import that .css file into our .html file.

How to import external stylesheet ?

|  |
| --- |
| **one.html:**  **<link rel="stylesheet" href="my-style.css">**  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"> |

“rel” means relationship type.

In external style sheet we don’t required to write <style> tag.



Inline style.

We write **style attribute** inside HTML tag/element.

Internal style.

We write **<style> tag**/**element** inside .html page.

External style.

We write **separate .css file** and we import that .ccs file into .html page.

**<link rel="stylesheet" href="my-style.css">**

RCA -> Root Cause Analysis. Finding out the root cause.

CSS code priority: Inline style **>** internal style **>** External style.

Some CSS properties: color, background-color, font-size, font-family, font-weight, text-align (Here alignment means position), <sub>, <sup>, <b>, <i>, <del> (strike the content),

Whatever the changes we are doing by HTML attributes, we can do by CSS properties also.

**Comments**: Gives additional information about our project.

HTML comments are not displayed in the browser, but they can help document your HTML source code.

We can use **CTRL + forward Slash (/)** buttons for commenting code in VSCode.

HTML Comments:

<!- - my HTML Comments here

-------

-- >

CSS Comments:

/\*

my HTML Comments here

my HTML Comments here

------

\*/

JavaScript Comments:

1. **//** double forward slash

**/\***

my HTML Comments here

my HTML Comments here

------

**\*/**

**Debugging**: Finding out the error which we got in our application, is call debugging. Line by line checking the code is called as debugging process.

Comments are also great for debugging HTML.

We can write inline comments also. Means inside tag also we can write the comments.

<p id=”empId” > This is My First **<!-- Paragraph -->** </p>

**Color:**

HTML colors are specified with predefined - color names, or with RGB, HEX, HSL, RGBA, or HSLA values.

Some CSS properties : Color**,** background-color**,** border (**border: 20px solid green;**),

How many ways we can represent colors in html : 5 ways we can represent.

Color formats : **Color name, RGB, HEX, hsl, rgba, hsla.**

Each parameter (red, green, and blue) defines the intensity of the color with a value between 0 and 255.

This means that there are 256 x 256 x 256 = 16777216 possible colors!

RGBA Color Values

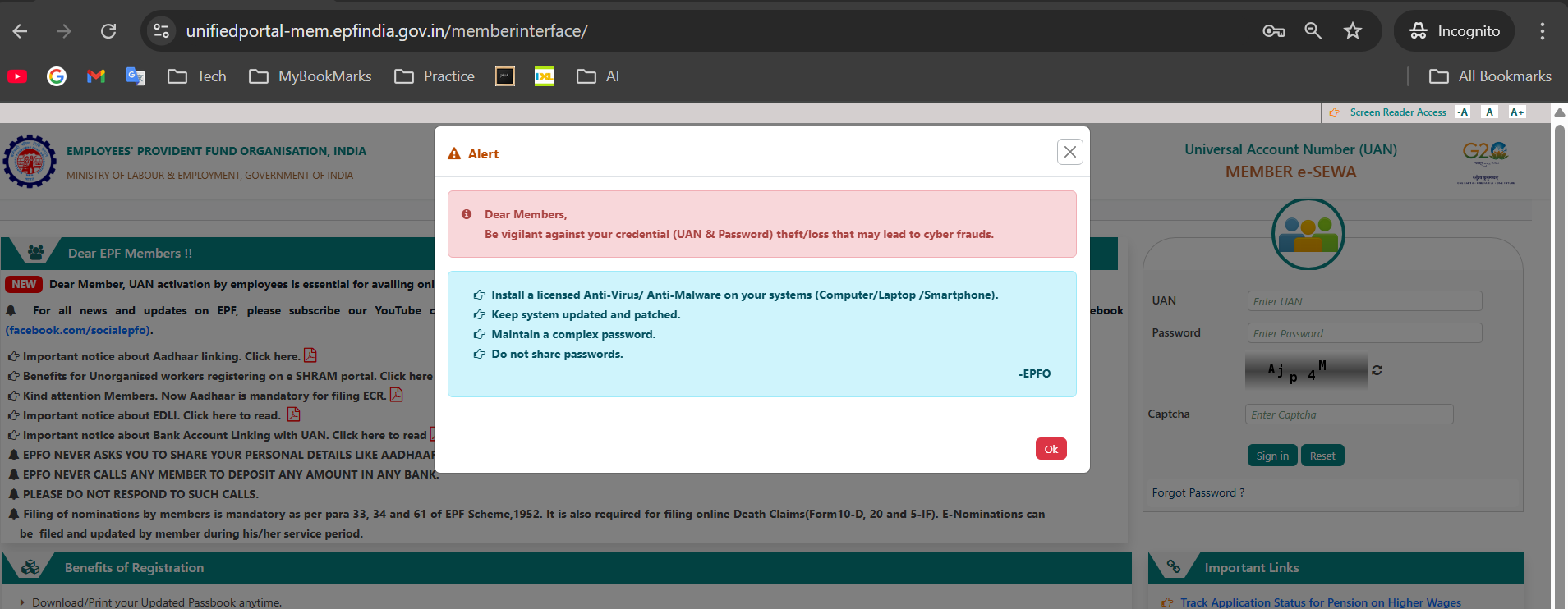
RGBA color values are an extension of RGB color values with an **Alpha channel** - which specifies the opacity for a color.

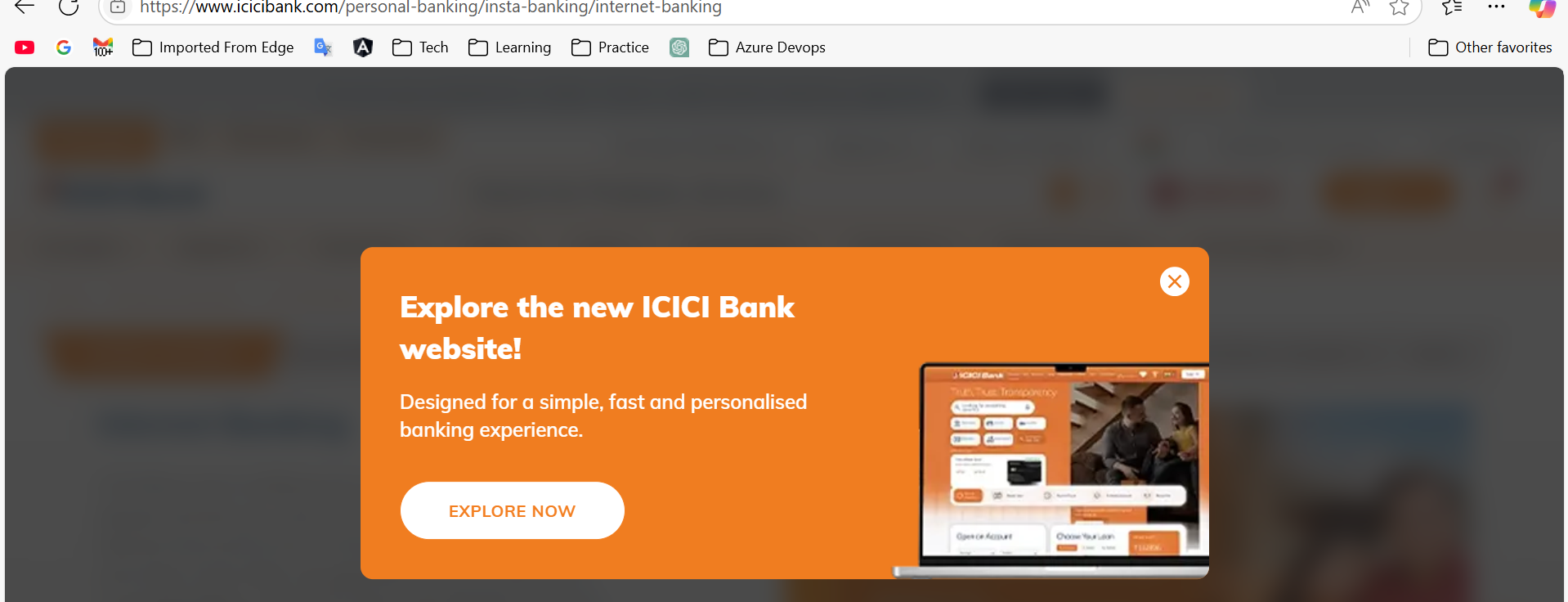
An RGBA color value is specified with:

**rgba(*red,* *green*, *blue, alpha*)**

**The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):**

**Opacity? 🡪 Means transparency.** Opacity is the degree to which content behind an element is hidden.





RGB rang is from 0 to 255.

Opacity range is from 0.0 to 1.0.

HEX Color Values: 16 digits.

**The 16 digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.**

0-9 = 10 values + A-F = 6 values = 16 values.

Combination of any two digits of 16 digits will form one color. Like this we need to prepare 3 sets of 2digits.

number to number combination is allowed and number to character combination allowed and character to character combination is also allowed.

#110034 🡪 Number to number combination.

#1ca6f8 🡪 Number, Character combination.

#FFaaBC🡪 Character to Character combination.

A hexadecimal color is specified with: **#RRGGBB**, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.

**To display black**, set all color parameters to 00, like this: **#000000**.

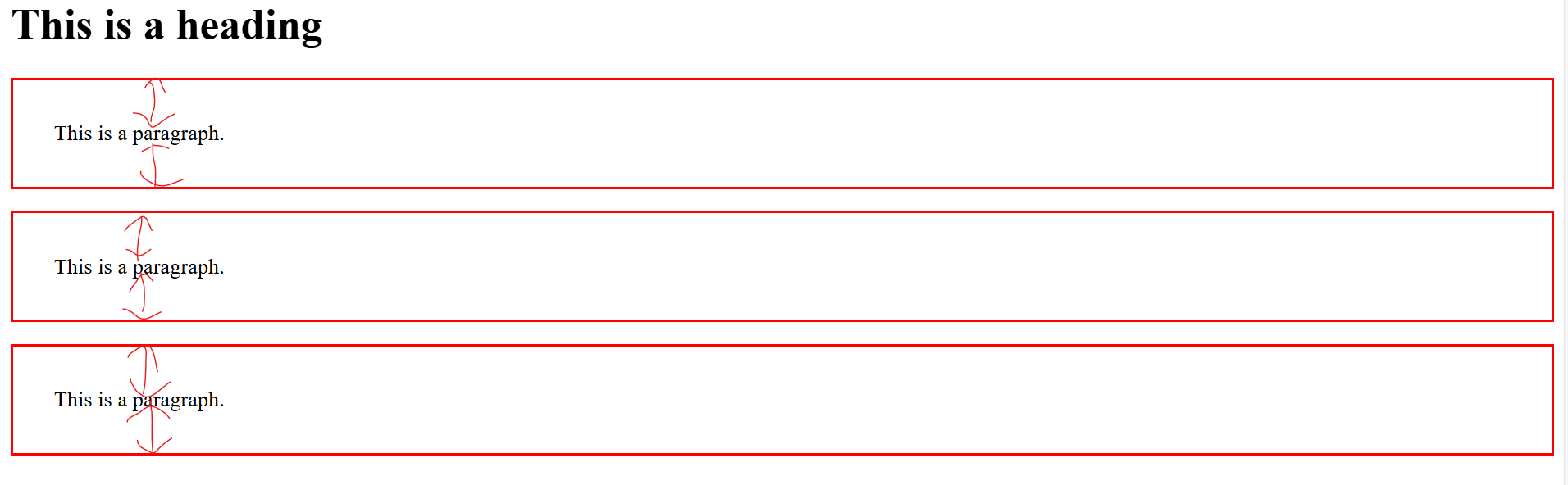
**To display white**, set all color parameters to ff, like this: **#ffffff**.

In real time projects developers always uses HEXa Decimal format only.

Cascading Style Sheets (CSS) is used to format the layout of a webpage.

 The word **cascading** means that a style applied to a parent element will also apply to all children elements within the parent.

The CSS **padding** property defines a padding **(space) between the text and the border.**



|  |
| --- |
| p {    border: 2px solid red;  **padding: 50px;**    } |

Please explain Input types. 🡪 text, radio, checkbox, file, submit, hidden, image, date, image ….etc.

Eg:

|  |
| --- |
| Enter User Name <input **type="text”** placeholder="Please enter your name" id=”nameId” class=”nameClass”/> |

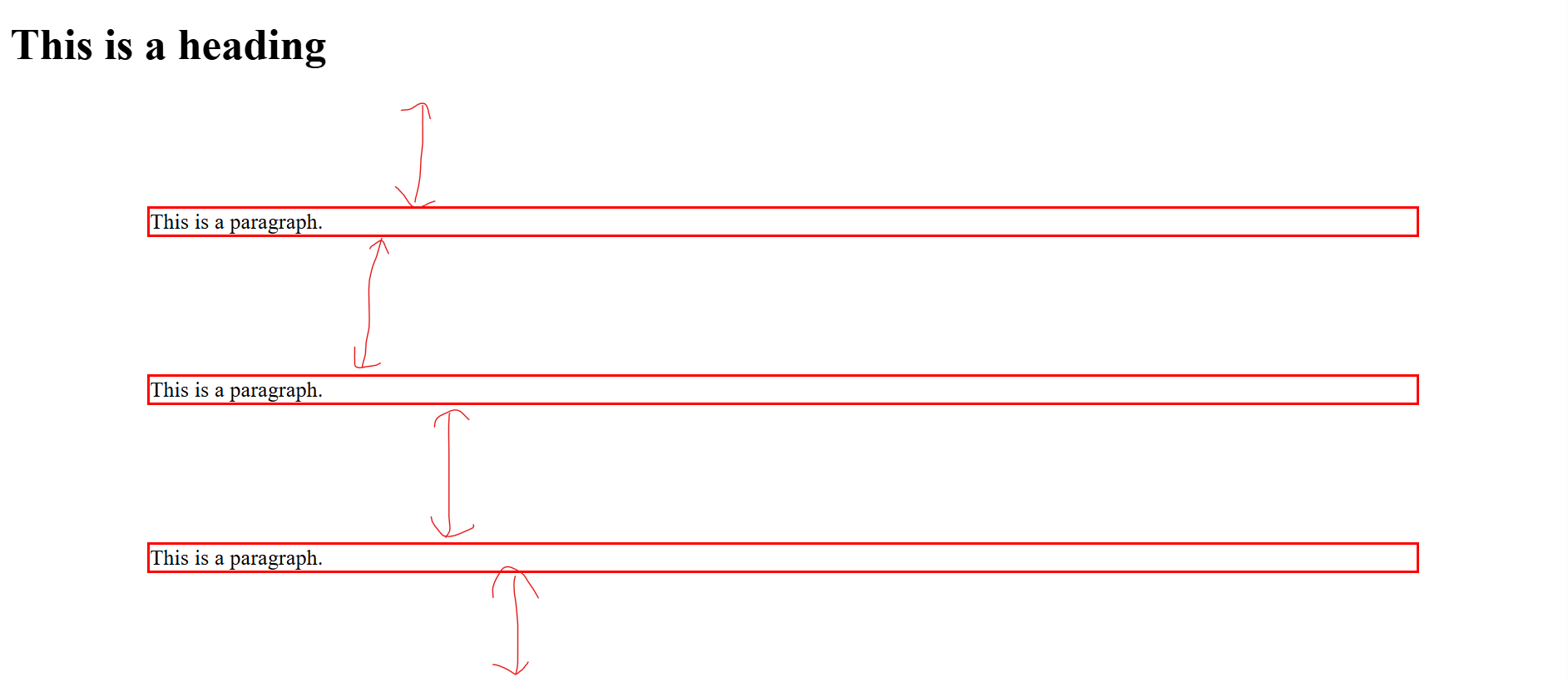
How to create Drop down in HTML;

|  |
| --- |
| **<select>**  **<option>**Select**</option>**    <option>Ap</option>    <option>TS</option>    <option>TN</option>    <option>KE</option>    <option>MH</option>  **</select>** |

**Margin**:

The CSS margin property defines a margin (space) outside the border.

|  |
| --- |
| p {    border: 2px solid red;  **margin: 100px;**  } |

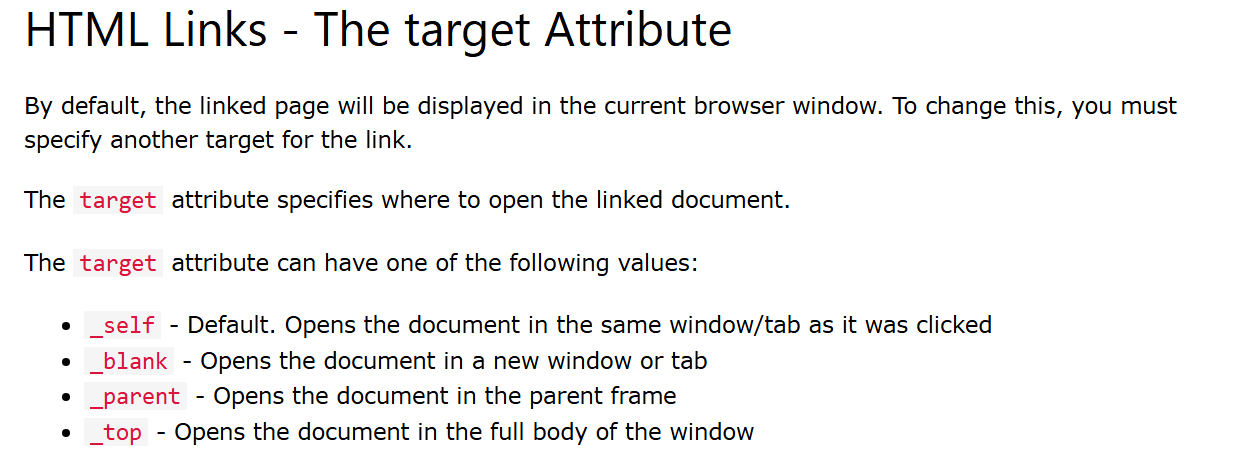


Use the CSS border property for borders

Use the CSS padding property for space inside the border

Use the CSS margin property for space outside the border

Links:



How to use image as hyperlink? 🡪 inside the <a> anchor tag we use <img> tag.

|  |
| --- |
| **<a** href="https://www.w3schools.com/html/html\_links.asp">  **<img** src="instagram3.ico"/>  **</a>** |

How to send mail using hyperlink?

|  |
| --- |
| <a href**="mailto:abc@gmail.com">**Send email</a> |

How to make button as hyperlink?

|  |
| --- |
| <button **onclick="document.location**='https://www.w3schools.com/'">HTML Tutorial</button> |

Explain below selectors:

|  |
| --- |
| <style>  **a:link {**              color: red;              background-color: yellow;              text-decoration: none;          }  **a:visited {**              color: pink;              background-color: transparent;              text-decoration: none;          }  **a:hover {**              color: rgb(0, 255, 187);              background-color: transparent;              text-decoration: underline;          }  **a:active {**              color: violet;              background-color: transparent;              text-decoration: underline;          }      </style> |

Content Jump: Bookmarking. Jumping from one location to another location within same html page.

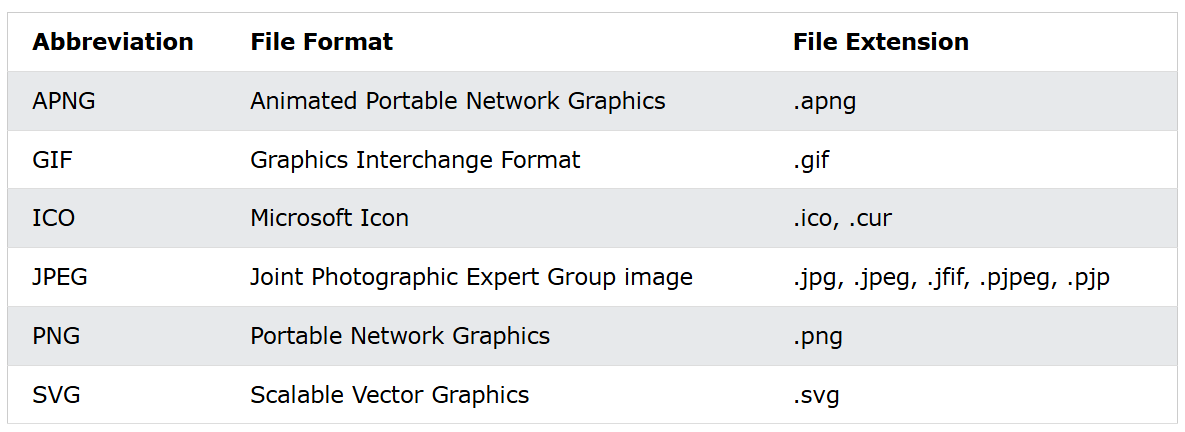
Mostly used for page indexing scenario.

Eg: PDF page index, jumps from index to particular page on our click action.

|  |
| --- |
| <p><a href**="#C4">**Jump to Chapter 4</a></p> 🡺 Find and Apply.  <p><a href**="#C10">**Jump to Chapter 10</a></p>  // 20 lines of code..  <h2 **id="C4"**>Chapter 4</h2>  <h2 **id="C10"**>Chapter 10</h2> |

**\*\*\*\*\*\*\*\*\*Here “#” represents selector “id” of the element(tag).**

**“#” means “id” attribute in any html element/tag.**



<img>

|  |
| --- |
| img src="img\_girl.jpg" alt="Girl in a jacket" style="width:500px;height:600px;"> |

Background Image:

|  |
| --- |
| <p style="**background-image: url('img\_girl.jpg')**;"> |

By default background image is repeated.

To avoid the background image from repeating itself, set the **background-repeat** property to **no-repeat**.

|  |
| --- |
| **test.html:**   body {              background-image: url('tux.gif');  **background-repeat: no-repeat;**          } |

If you want the background image to cover the entire element, you can set the **background-size** property to **cover**.

Also, to make sure the entire element is always covered, set the background-attachment property to fixed:

|  |
| --- |
| **test.html:**  <style> body {   background-image: url('img\_girl.jpg');   background-repeat: no-repeat; **background-attachment: fixed;   background-size: cover;** } </style> |

If you want the background image to stretch to fit the entire element, you can set the background-size property to 100% 100%:

|  |
| --- |
| **test.html:**  <style> body {   background-image: url('img\_girl.jpg');   background-repeat: no-repeat;   background-attachment: fixed; **background-size: 100% 100%;** } </style> |

**Favicon:**

A favicon is a small image displayed next to the page title in the browser tab.

Use the HTML **<link>** element to insert a favicon

|  |
| --- |
| <head>   <title>My Page Title</title> **<link rel="icon" type="image/x-icon" href="/images/favicon.ico">** </head> |

**<title> tag/element: The title is shown in the browser's title bar**

|  |
| --- |
| <head> **<title>**HTML Tutorial**</title>** </head> |

**Tables:**

HTML tables allow web developers to arrange data into rows and columns.

We will represent table with **<table>** element / tag.

Rows?: Horizontal representation is called as Row. 🡺 <tr>

Columns?: Vertical representation is called as column. 🡺 <th> or <td>

Cell: single box in each row is called as cell. 🡺 <th> or <td>

Table heading? 🡪 column heading names. 🡺 <th>

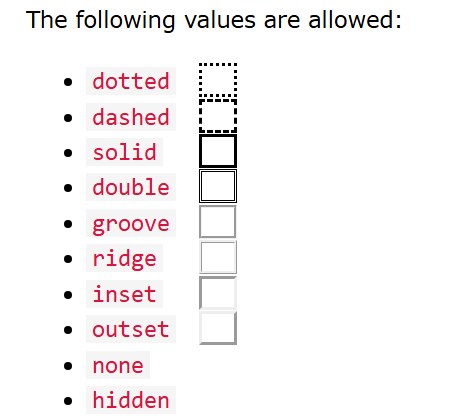
Table data? 🡪 actual data in cell is called as table data. Each cell in row is represented as table data. 🡺 <td>

A table in HTML consists of table cells inside rows and columns.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>      <style>  **table {**  **font-family: arial, sans-serif;**  **border-collapse: collapse;**  **width: 100%;**  **}**  **td,**  **th {**  **border: 1px solid #10cf93;**  **text-align: left;**  **}**      </style>  </head>  <body>  **<table>**  **<tr>**  **<th>**Name</th>              <th>age</th>              <th>Cell</th>              <th>City</th>          </tr>  **<tr>**  **<td>**Azad</td>              <td>21</td>              <td>343434</td>              <td>MVG</td>          </tr>          <tr>              <td>Arafath</td>              <td>21</td>              <td>67677</td>              <td>YPl</td>          </tr>      </table>  </body>  </html> |

**{ border: 1px solid red;}**

So far we wrting borer style as “solid” only, but we are having multiple options like below.



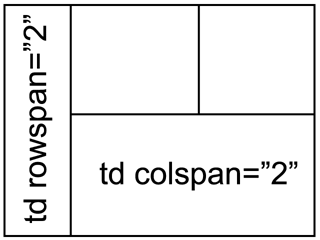
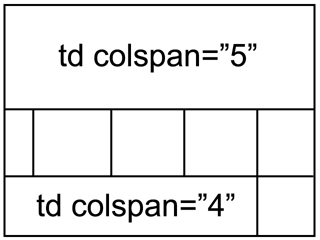
|  |
| --- |
| table {  Border: 1px **dashed** red;  } |

Use the **style** attribute with the **width** or **height** properties to specify the size of a table, row or column.

**Colspan:**

Header for Multiple Columns. We can merge column cells by using “**colspan**” attribute. **Colspan count tells how many cells you want to merge.**

|  |
| --- |
| <table style="width:100%">    <tr>      <th **colspan="3">**Employees Table</th>    </tr>    <tr>      <td>Kamal</td>      <td>Shaik</td>      <td>36</td>    </tr>    <tr>      <td>Azad</td>      <td>Shaik</td>      <td>23</td>    </tr>    <tr>      <td>Arafath</td>      <td>Md</td>      <td>23</td>    </tr>  </table> |

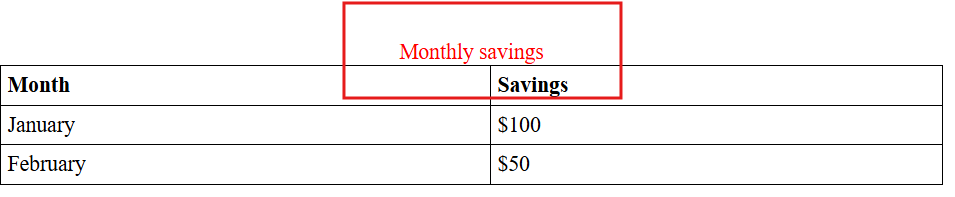
**Colspan 🡪 Merging cells from multiple columns.**

**Rowspan 🡪 Merging cells from multiple rows.**

Here 2 or 4 or 5. means cells count.

**Table Caption:**

|  |
| --- |
| <caption style="color: red;">Monthly savings</caption> |



**Padding? 🡪**

By default padding css property **{padding : 20px}** takes all sides of content. Means it is by default

Padding-top, padding-right, padding-bottom, padding-left.

If you want different values for four sides, we use above css properties.

|  |  |
| --- | --- |
| **Approach-1**  table, th, td {    border: 1px solid black;    border-collapse: collapse;    color: green;  }  th, td {  **padding:50px 60px 70px 80px;**    text-align: left;    font-family: sans-serif;  }  Here values count in **CLOCK-WISE format**, means first values it tales from padding-top next padding-right, next padding-bottom next padding-left this is the order of applying. | **Approach-2**  table, th, td {    border: 1px solid black;    border-collapse: collapse;    color: green;  }  th, td {    /\* padding:50px 60px 70px 80px; \*/    **padding-bottom: 40px;**  **padding-bottom: 50;**  **padding-top: 20px;**    text-align: left;    font-family: sans-serif;  **padding-right: 30px;**  } |

HTML Table Styling:

By using CSS you will get a nice zebra stripes effect.

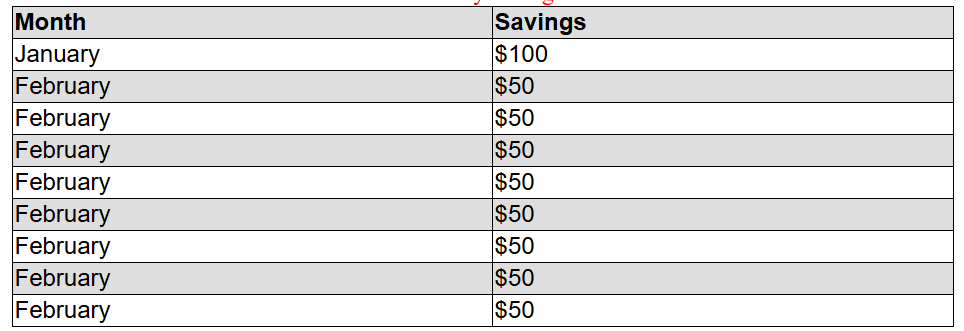
We can apply colors to rows as even and odd numbers.

Odd means 🡪 1,3,5,7,9,…etc.

Even means 🡪 2,4,6,8,10,…etc.

|  |  |
| --- | --- |
| **tr:nth-child(even) {**   background-color: #D6EEEE; } | **tr:nth-child(odd) {**   background-color: #D6EEEE; } |

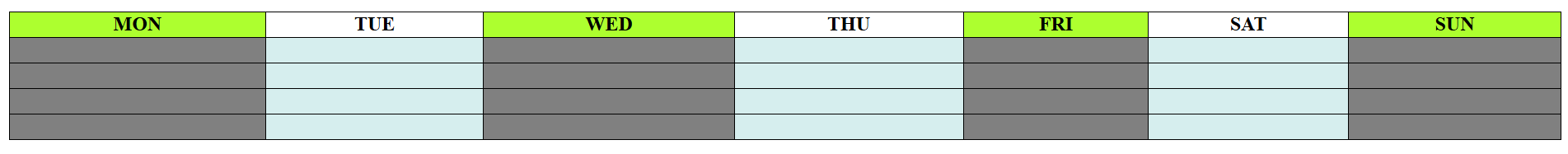
We can see output as like below.



For applying **CSS colors to vertically arranged columns** we need to use below css code.

|  |  |
| --- | --- |
| **td:nth-child(even) {**    background-color: #D6EEEE;  }  **td:nth-child(odd){**    background-color: grey;  }  **th:nth-child(odd){**    background-color: greenyellow;  } | **td:nth-child(odd) {**    background-color: #D6EEEE;  }  **td:nth-child(odd){**    background-color: grey;  }  **th:nth-child(even){**    background-color: greenyellow;  } |

We can see output as like below.



We can use Alpha values in RGBA color format, and we can additionally pass alpha transparency values also.

|  |
| --- |
| tr:nth-child(even) {    background-color: **rgba**(150, 212, 212**, 0.4**);  }  th:nth-child(even), td:nth-child(even) {    background-color: **rgba**(150, 212, 212**, 0.4);**  }  </st |

Hoverable Table:

Use the **:hover** selector on **tr** to highlight table rows on mouse over:

|  |
| --- |
| **tr:hover** {  background-color: #D6EEEE;  } |

So finally …

for

Rows 🡪 tr:nth-child(even/odd)

Columns -> th:nth-child(even/odd) , td:nth-child(even/odd)

For hovering -> tr:hover

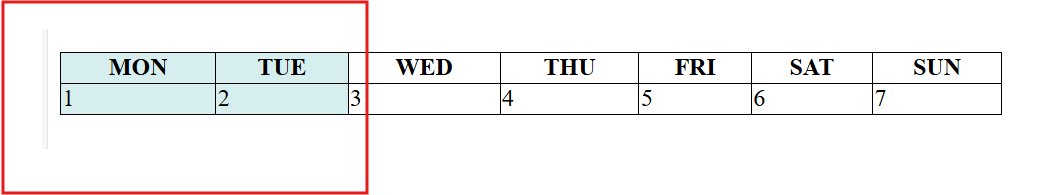
**HTML Table Colgroup:**

The <colgroup> element is used to style specific columns of a table.

If you want to style the first two columns of a table, use the <colgroup> and <col> elements.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  table, th, td {  border: 1px solid black;  border-collapse: collapse;  }  </style>  </head>  <body>  <h2>Colgroup</h2>  <p>Add the a colgroup with a col element that spans over two columns to define a style for the two columns:</p>  <table style="width: 100%;">  **<colgroup>**  **<col span="2" style="background-color: #D6EEEE">**  **</colgroup>**  <tr>  <th>MON</th> <th>TUE</th><th>WED</th><th>THU</th><th>FRI</th><th>SAT</th><th>SUN</th>  </tr>  <tr>  <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td>  </tr>  </table>  </body>  </html> |

Output will be like below. Only two columns are highlighted with color.



Examples:

|  |
| --- |
| <colgroup>  <col span="2" style="background-color: #D6EEEE">  <col span="3" style="background-color: pink">  </colgroup>  Eg2:  <colgroup>  <col span="3">  <col span="2" style="background-color: pink">  </colgroup>  Eg: 3    <colgroup>     <col span="2">     <col span="3" style="visibility: collapse">   </colgroup> |

**HTML Lists:**

HTML lists allow web developers to group a set of related items in lists.

List :

1. Unordered HTML list 🡪 <ul> 🡪 <li> 🡪 black circles / bullets are default.
2. Ordered HTML list 🡪 <ol> 🡪 <li> 🡪 numbers are default

UL:

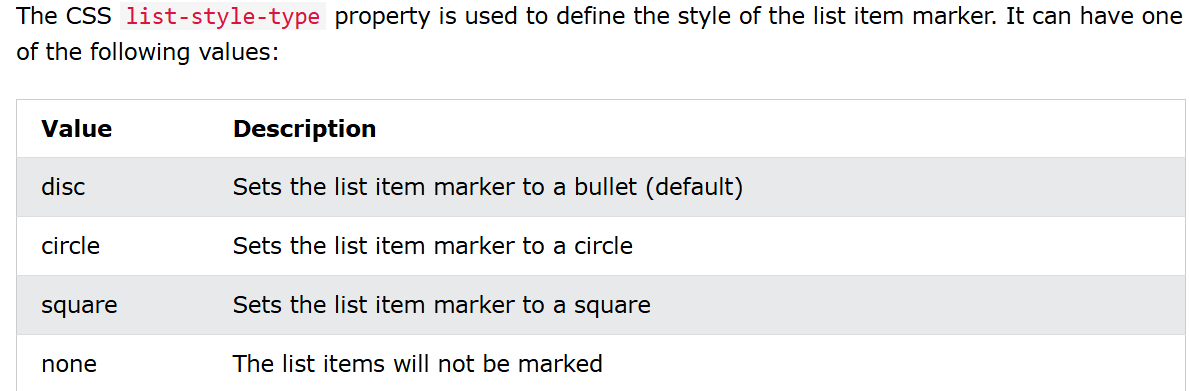
An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

OL:

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default:



|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <h2>Unordered List with Disc Bullets</h2>  <**ul style="list-style-type:square;">**    <li>Coffee</li>    <li>Tea</li>    <li>Milk</li>  </ul>  </body>  </html> |

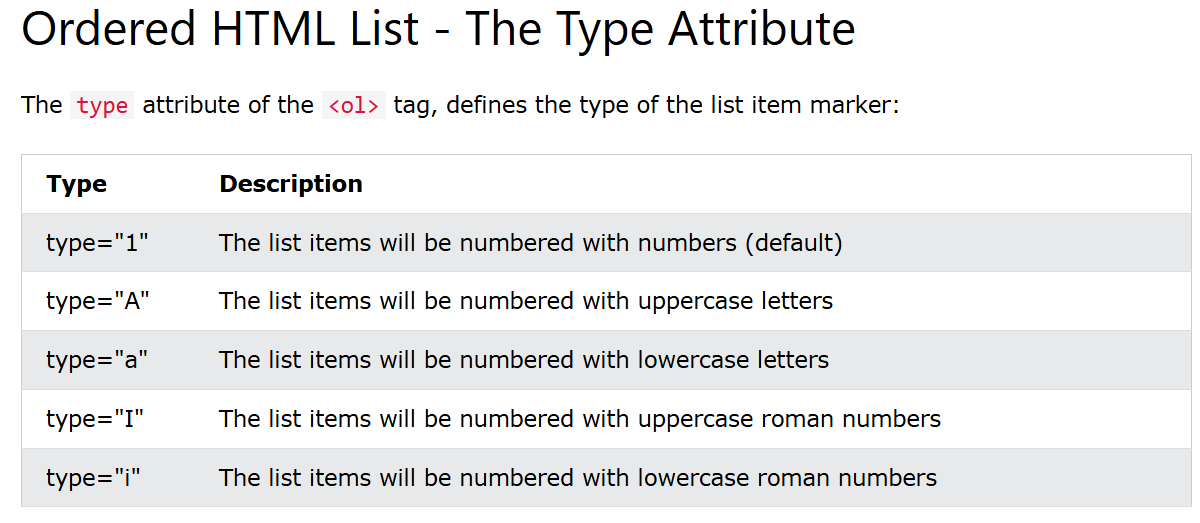
Nested HTML Lists

Lists can be nested (list inside list).

|  |
| --- |
| <ul>    <li>Coffee</li>    <li>Tea      <ul>        <li>Black tea</li>        <li>Green tea</li>      </ul>    </li>    <li>Milk</li>  </ul> |

* Use the HTML <ul> element to define an unordered list
* Use the CSS list-style-type property to define the list item marker
* Use the HTML <li> element to define a list item
* Lists can be nested
* List items can contain other HTML elements
* **Use the CSS property float:left to display a list horizontally**

|  |
| --- |
| li {  **float: left;**  } |



|  |
| --- |
| <ol **type="I">**      <li><a href="#home">Home</a></li>      <li><a href="#news">News</a></li>      <li><a href="#contact">Contact</a></li>      <li><a href="#about">Login</a></li>    </ol> |

Control List Counting

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the start attribute:

|  |
| --- |
| <ol **start="50"**>   <li>Coffee</li>   <li>Tea</li>   <li>Milk</li> </ol> |

**ASCII** stands for American Standard Code for Information Interchange.

<https://www.ascii-code.com/>

<https://www.w3schools.com/charsets/ref_html_ascii.asp>

**HTML id Attribute:**

|  |  |
| --- | --- |
| In HTML:  <p **id="i1" class="employee"**>This is paragraph......</p> | Test.css:  **#i1** {  background-color: red;  }  **.empoyee** {  color:grey;  } |

The HTML id attribute is used to specify a **unique** id for an HTML element.

You cannot have more than one element with the same id in an HTML document.

For example, student id, employee id, hall ticket number… always these should be unique.

No two students should have same id/ hall ticket number, no two employees have same employee id.

**This “id” attribute is used in CSS coding and JavaScript coding also.**

**This “class” attribute is used in CSS coding and JavaScript coding also.**

The syntax for id is: write a hash character (#), followed by an id name. Then, define the CSS properties within curly braces {}.

**Note:** The id name is case sensitive!

Difference Between Class and ID

A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page.

**Bookmark?** 🡪 Jumping from one area to another area within same browser page. It will be helpful if your web page is very lengthy. It will avoid scrolling in lengthy page.

Eg: <https://en.wikipedia.org/wiki/India#Visual_art>

**HTML Bookmarks with ID and Links:**

HTML bookmarks are used to allow readers to jump to specific parts of a webpage.

Bookmarks can be useful if your page is very long.

To use a bookmark, you must first create it, and then add a link to it.

Then, when the link is clicked, the page will scroll to the location with the bookmark.

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  **<p><a href="#C4">Jump to Chapter 4</a></p>**  **<p><a href="#C10">Jump to Chapter 10</a></p>**  <h2>Chapter 1</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 2</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 3</h2>  <p>This chapter explains ba bla bla</p>  <h2 **id="C4">**Chapter 4</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 5</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 6</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 7</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 8</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 9</h2>  <p>This chapter explains ba bla bla</p>  <h2 **id="C10">**Chapter 10</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 11</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 12</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 13</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 14</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 15</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 16</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 17</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 18</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 19</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 20</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 21</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 22</h2>  <p>This chapter explains ba bla bla</p>  <h2>Chapter 23</h2>  <p>This chapter explains ba bla bla</p>  </body>  </html> |

**HTML “class” Attribute:**

The HTML class attribute is used to specify a class for an HTML element.

Multiple HTML elements can share the same class.

**Tip:** The class attribute can be used on **any** HTML element.

**Note:** The class name is case sensitive!

|  |
| --- |
| <h2 **class="city">**London</h2> |

|  |
| --- |
| **.city {**   background-color: tomato;   color: white;   padding: 10px; } |

**Multiple Classes:**

HTML elements can belong to more than one class. **We can mention multiple class names in same class attribute**.

To define multiple classes, separate the class names with a space, e.g. <**div class="city main">.** The element will be styled according to all the classes specified.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  **.city** {  background-color: tomato;  color: white;  padding: 10px;  }  **.main** {  text-align: center;  }  </style>  </head>  <body>    <h2 **class="city main">**London</h2>  <h2 class="city">Paris</h2>  <h2 class="city">Tokyo</h2>  </body>  </html> |

* The HTML class attribute specifies one or more class names for an element
* Classes are used by CSS[.className] and JavaScript(getElementsByClassName()) to select and access specific elements
* The class attribute can be used on any HTML element
* The class name is case sensitive
* Different HTML elements can point to the same class name
* JavaScript can access elements with a specific class name with the getElementsByClassName() method

**HTML Div Element:**

The <div> element is used as a container for other HTML elements.

**The <div> element is by default a block (occupy) element**, meaning that it takes all available width, and comes with line breaks before and after.

**The <div> element is often used to group sections of a web page together.**

|  |
| --- |
| Lorem Ipsum <**div>**I am a div**</div>** dolor sit amet. |

|  |
| --- |
| <!DOCTYPE html>  <html>  <style>  div {      border: 1px solid red;    width: 363px;    float: left;    padding: 23px;  }  </style>  <body>  <h1>Center align a DIV element</h1>  <div style="background-color: green;">    <h2>London</h2>    <p>London is the capital city of England.</p>    <p>London has over 9 million inhabitants.</p>  </div>  <div style="background-color: greenyellow;">    <h2>Inda</h2>    <p>London is the capital city of England.</p>    <p>London has over 9 million inhabitants.</p>  </div>  <div style="background-color: violet;">    <h2>USA</h2>    <p>London is the capital city of England.</p>  <p>London has over 9 million inhabitants.</p>  </div>  <div style="background-color: grey;">    <h2>AUS</h2>    <p>London is the capital city of England.</p>  <p>London has over 9 million inhabitants.</p>  </div>  </body>  </html> |

Aligning <div> elements side by side🡪 **float: left**

When building web pages, you often want to have two or more <div> elements side by side.

**Float**

The CSS float property was not originally meant to align <div> elements side-by-side, but has been used for this purpose for many years.

The CSS float property is used for **positioning and formatting content** and allows elements to be positioned horizontally, rather than vertically.

**Inline-block:**

If you change the <div> element's display property from block **to inline-block**, the <div> elements will no longer add a line break before and after, and will be displayed side by side instead of on top of each other.

**Default nature of block level (<div> or <p> …etc) elements:**

1. **It will block the entire width.**
2. **It will add line breaks before and after the <div> element.**

**Eg:**

|  |
| --- |
| **Test.html:**  div {      background-color: red;    }  AAAAAAAAAAAA<div>BBBBBBBBBBBBBBBBBB</div>CCCCCCCCCCCCCC |

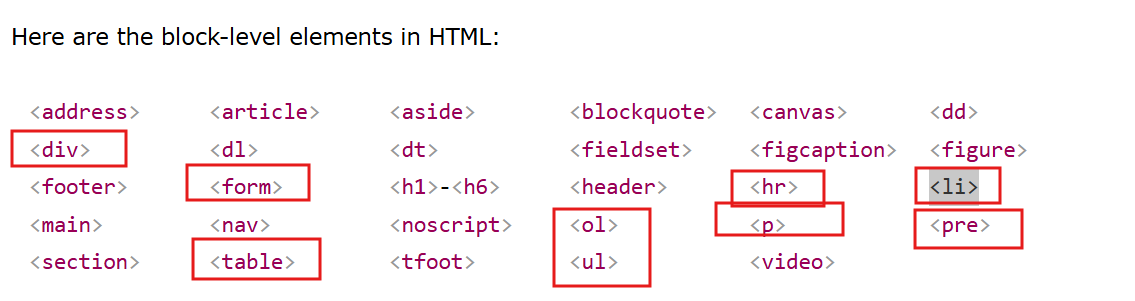
**{display: “inline-block”} is used to over come above problem**.

|  |
| --- |
| div {      background-color: red;  **display: inline-block;**    } |

**HTML Block and Inline Elements:**

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

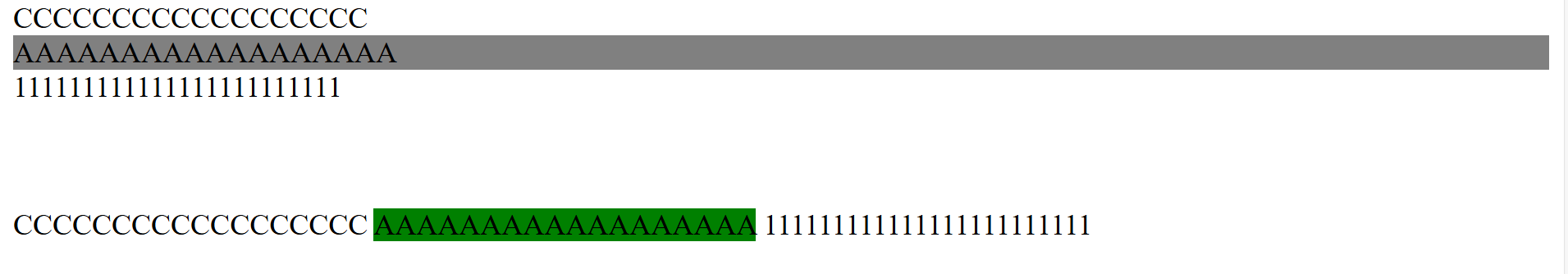
A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

**Two commonly used block elements are: <p> and <div>.**

**Inline Elements:**

**Elements are 2 types.**

1. **Block level elements. --> Blocks entire width, add line breaks before and after the element.**
2. **Inline elements. --> Not Blocks entire width, Not add line breaks before and after the element.**



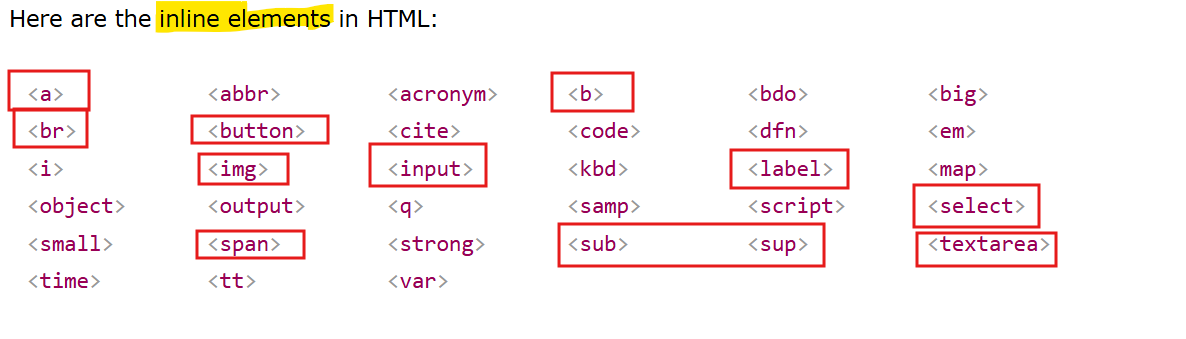
Html code:

|  |
| --- |
| <body>    CCCCCCCCCCCCCCCCCC  **<div>**AAAAAAAAAAAAAAAAAA**</div>** 111111111111111111111111  <br><br>  <br><br>  CCCCCCCCCCCCCCCCCC  **<span>**AAAAAAAAAAAAAAAAAA**</span>** 111111111111111111111111  </body> |

**Inline Elements:**

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.



**Note:** An inline element cannot contain a block-level element!

* A block-level element always starts on a new line and takes up the full width available
* An inline element does not start on a new line and it only takes up as much width as necessary
* The <div> element is a block-level element and is often used as a container for other HTML elements
* The <span> element is an inline container used to mark up a part of a text, or a part of a document

**HTML Forms:**

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

Elements used so far:

<h1> ..<h6>, <div> , <p>, <span>, <table> <th>, <tr>, <td>, <ul>,<ol>,<li>, <a>, <img>, <br>,

**The <form> Element:**

The HTML **<form>** element is used to create an HTML form for **user input**.

The **<form>** element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

|  |
| --- |
| <form> . *form elements all input fields comes here.* . </form> |

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  **<form action="requestPath / serverPath/">**     Enter Name**: <input type="text" id="fname" name="fname">**<br>    Enter password: **<input type="text" id="lname" name="lname"><**br><br>  **<input type="submit" value="Login">**  **</form>**   </body>  </html> |

**\*\*\*\* submit button always looks for “action” attribute inside the <form> element. It will take user entered input data to server.**

Submit button we can write in three ways.

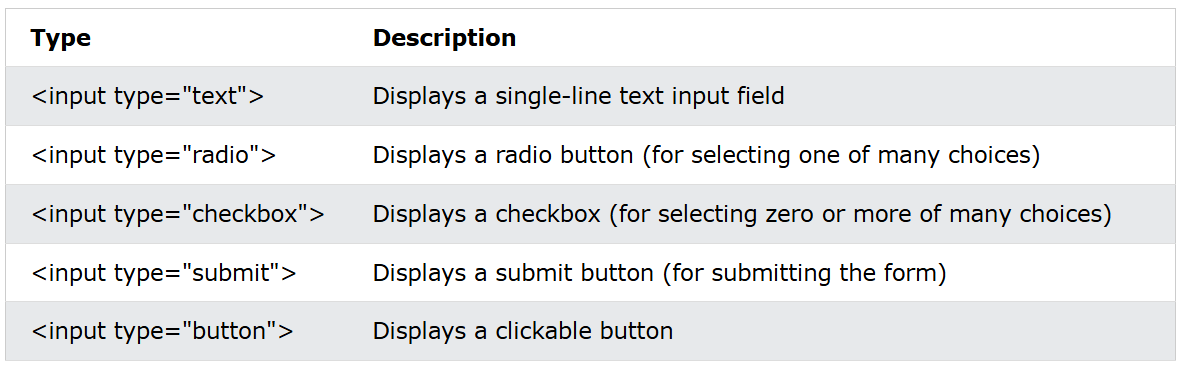
1. <input type=”submit”/>
2. <input type=”button”/>
3. <button …/>

|  |
| --- |
| <input type="submit" value="Login" />  <input type="button" value="Login" />  <button>Click Me...</button> |

**The <input> Element:**

The HTML <input> element is the most used <form> element.

An **<input>** element can be displayed in many ways, depending on the **type** attribute.



|  |
| --- |
| <input type="button">  <input type="checkbox"> <br>  <input type="color"><br> -🡪 color picker  <input type="date"><br> 🡪 date picker.  <input type="datetime-local"><br> 🡪 date picker with time stamp.  <input type="email"><br>  <input type="file"><br>  <input type="hidden"><br>  <input type="image"><br>  <input type="month"><br>  <input type="number"><br>  <input type="password"><br>  <input type="radio"><br>  <input type="range"><br>  <input type="reset"><br>  <input type="search"><br>  <input type="submit"><br>  <input type="tel"><br>  <input type="text"><br>  <input type="time"><br>  <input type="url"><br>  <input type="week"><br> |

|  |
| --- |
| <form>  **<label for="fname">First name:</label**><br>    <input type="text" **id="fname**" name="fname" value="John"><br>      <label **for="lname**">Last name:</label><br>    <input type="text" **id="lname**" name="lname" value="Doe">  </form> |

**The <label> Element:**

The <label> tag defines a label for many form elements.

The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element.

The **for** **attribute** of the **<label> tag** should be equal to the **id attribute** of the **<input>** element to bind them together.

**Radio Buttons:**

Radio buttons let a user **select ONE** of a limited number of choices.

By default all radio buttons are allowed to select, but if you want only one radio button to select, we need to write “**name**” attribute and it should have same value for all radio button elements.

|  |
| --- |
| Select Your Mother language:      <form>          English <input **type="radio"** **name="gender"** \>          Telugu<input type="radio" **name**="**gender**" \>          Tamil<input type="radio" **name**="**gender**" \>          Malayalam<input type="radio" **name**="**gender**" \>          Kannada<input type="radio" **name**="**gender**" \>          Hindi<input type="radio" **name**="**gender**" \>      </form> |

**Checkboxes:**

Checkboxes let a user **select ZERO or MORE** options of a limited number of choices.

|  |
| --- |
| <form **action=”serverPath”>**          English <input type="checkbox" \>          Telugu<input type="checkbox" \>          Tamil<input type="checkbox" \>          Malayalam<input type="checkbox" \>          Kannada<input type="checkbox"  \>          Hindi<input type="checkbox"\>      </form> |

**HTML Form Attributes:**

**The Action Attribute**

The action attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

In the example below, the form data is sent to a file called "action\_page.php". This file contains a server-side script that handles the form data

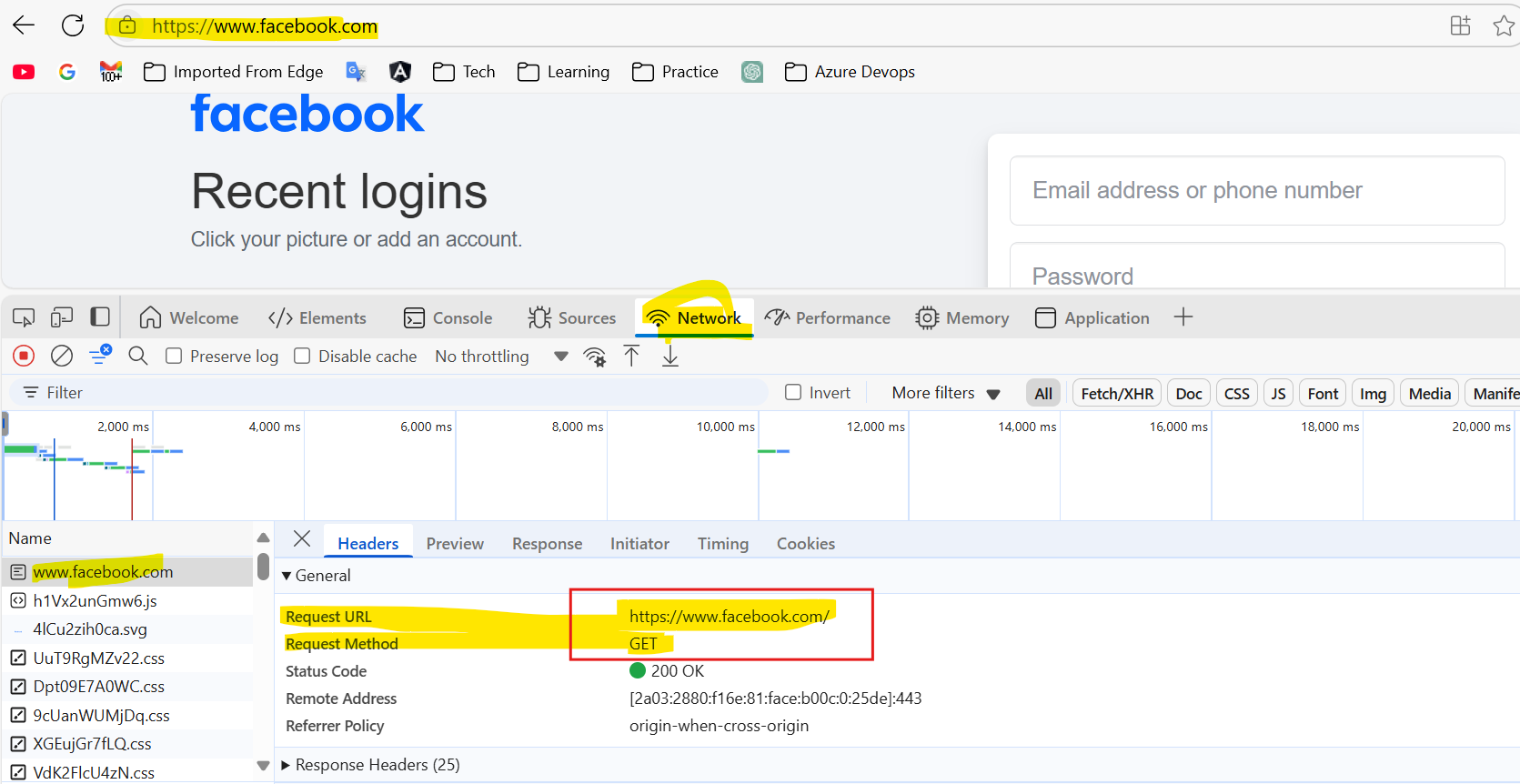
**Tip:** If we don’t write the action attribute in <form> element, on click of the submit button the action is set to the current page.

**The Target Attribute:**

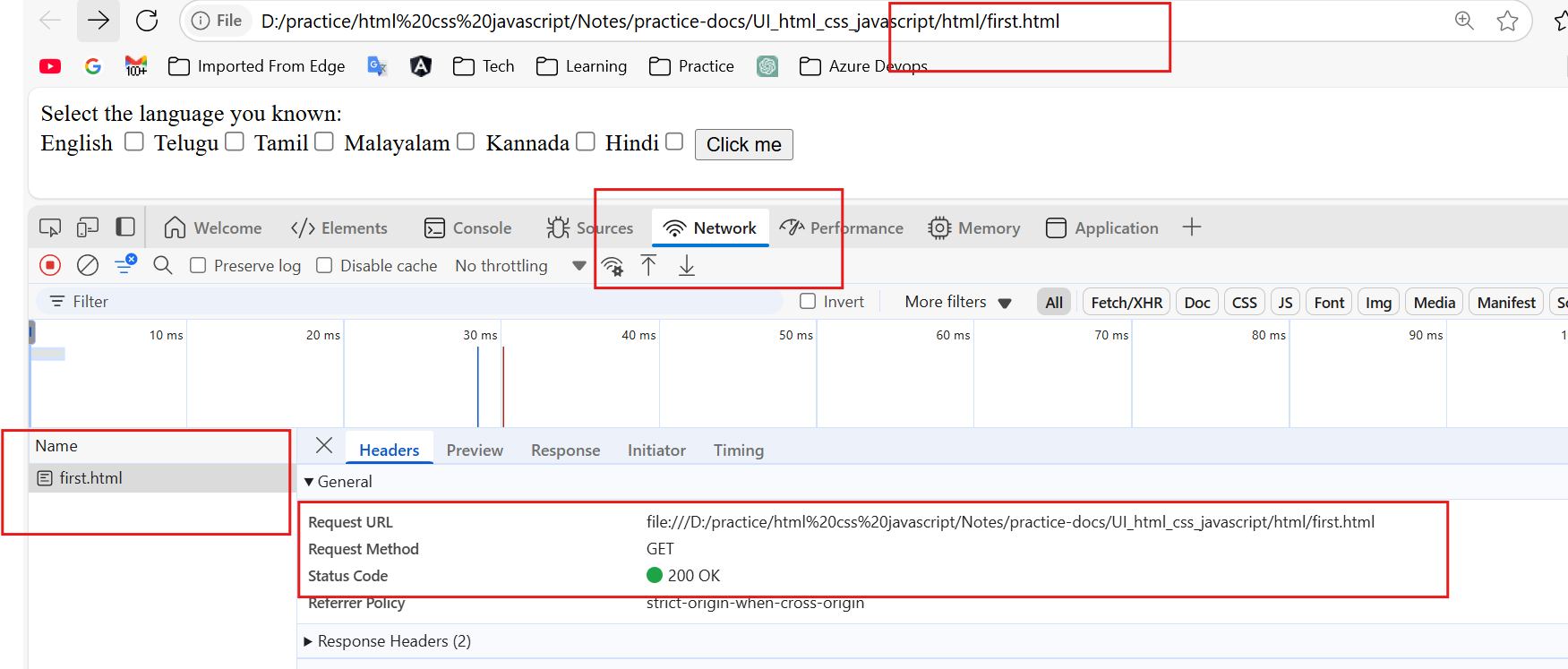
**The Method Attribute:**

This tells what is the type of form submission.

**GET\* (Fetch details), POST (create NEW details and sending to server ), PUT (Update existing details and sending to server-EDIT mode),** **DELETE**, PATCH, HEAD, OPTIONS, and CONNECT.



Our page:



By default all web pages have “Request Method” type. “GET” is default request method.

<https://www.w3schools.com/tags/ref_httpmethods.asp>

The method attribute specifies the HTTP method to be used when submitting the form data.

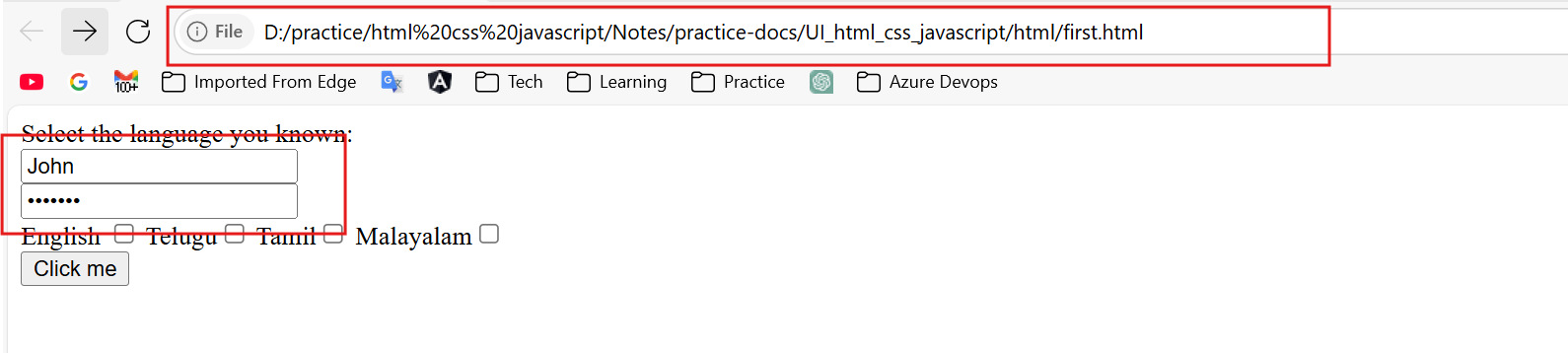
The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

HTTP – it is a protocol for sending the request, protocol means rules and regulations, while sending request to server, always it needs one rotocol.

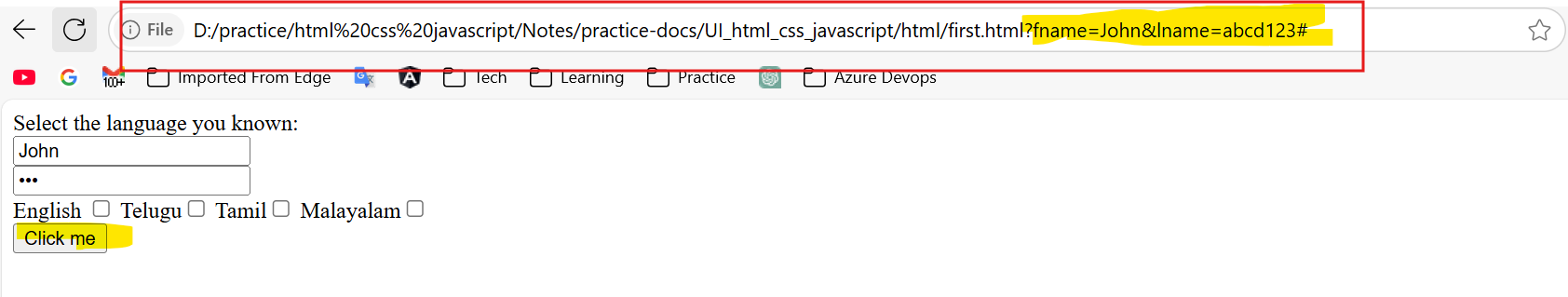
By using GET request method also we can send the data, but data is not secured, what ever the data we filled that will be visible on URL bar.

|  |
| --- |
| <**form** action="#" **method="GET">**          <input type="text" id="fname" name="fname" value="John"><br>          <input type="password" id="lname" name="lname" value="Doe"><br>          English <input type="checkbox" \>          Telugu<input type="checkbox" \>          Tamil<input type="checkbox" \>          Malayalam<input type="checkbox" \>          <input **type="submit**" value="Click me">      </**form**> |

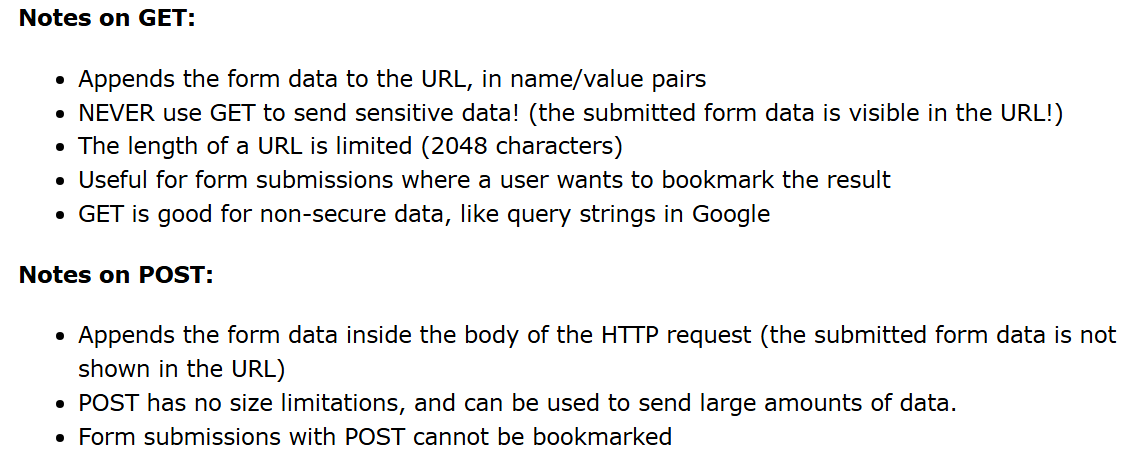
Before submitting the form we can see below data and URL bar.



After submitting the form we can see below data and URL bar.



**For form submission GET is not recommended, as it is not secured. GET always used to get the details from server.**



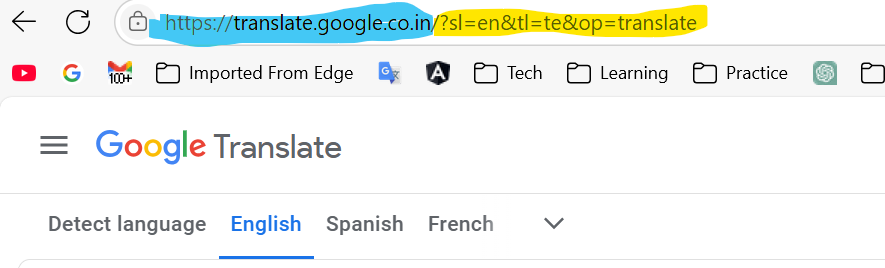
Append means 🡪 adding data at the end.

1MB -🡪 1024 KB.

2MB 🡪 2048 KB.

URL Appended data is called as “**query String**”.

Its format is like “serverPath**?key1=value1&key2=value2&key3=value3&……………**”



**Tip:** Always use POST if the form data contains sensitive or personal information!

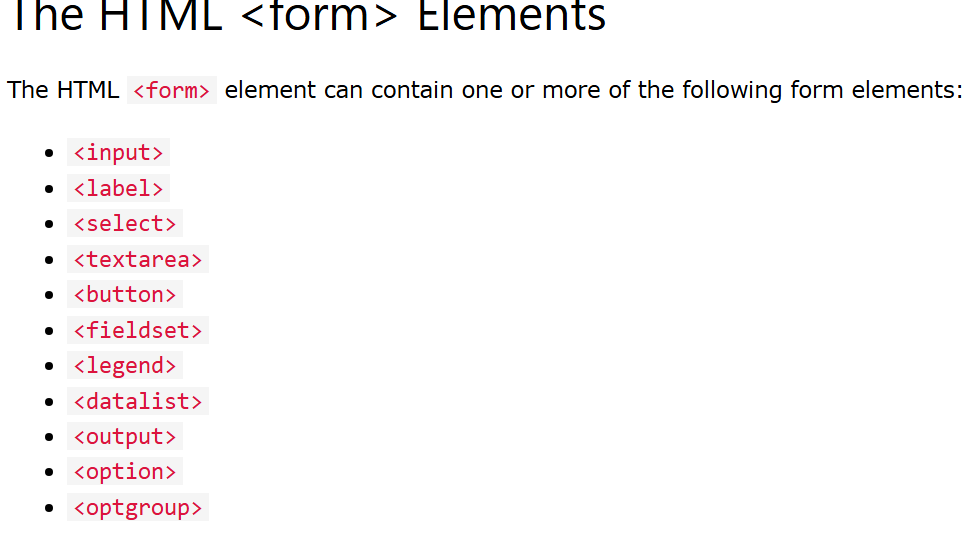
**The Autocomplete Attribute:**

**The Novalidate Attribute:**

The **novalidate** attribute is a **boolean** attribute.

When present, it specifies that the form-data (input) should not be validated when submitted.

|  |
| --- |
| <form action="/action\_page.php" **novalidate**> |



**The <select> Element:**

The <select> element defines a drop-down list:

|  |
| --- |
| <form action="/action\_page.php">  <label for="cars">Choose a car:</label>  **<select id="cars" name="cars">**  **<option value="volvo">Volvo</option>**  <option value="saab">Saab</option>  <option value="fiat">Fiat</option>  <option value="audi">Audi</option>  </select>  <input type="submit">  </form> |

By default, the first item in the drop-down list is selected.

To define a **pre-selected option**, add the “**selected”** **attribute** to the option:

|  |
| --- |
| <select>      <option value="000">Please select</option>      <option value="001">English</option>  **<option value="002" selected>**Telugu</option>  **<option value="003" disabled>**Tamil</option>    </select> |

**“disabled”** **attribute** will be used to disable the element. We can use this “disabled” attribute to all types of input elements to disable.

<select> tag Visible Values:

Use the **size attribute** to specify the number of visible values:

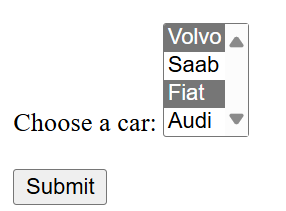
|  |
| --- |
| <form action="/action\_page.php">    <label for="cars">Choose a car:</label>    <select id="cars" name="cars" **size="4">**      <option value="volvo">Volvo</option>      <option value="saab">Saab</option>      <option value="fiat">Fiat</option>      <option value="audi">Audi</option>    </select><br><br>    <input type="submit">  </form> |

|  |  |
| --- | --- |
| **Before adding “size” attribute** | **After adding “size” attribute** |

**Allow Multiple Selections:**

Use the **multiple** **attribute** to allow the user to select more than one value:

|  |
| --- |
| <select id="cars" name="cars**" size="4" multiple**>   <option value="volvo">Volvo</option>   <option value="saab">Saab</option>   <option value="fiat">Fiat</option>   <option value="audi">Audi</option> </select> |



**The <textarea> Element:**

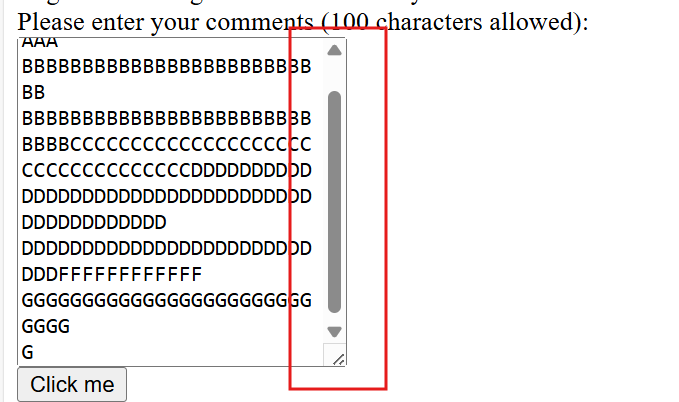
It will be mostly used for comment section.

|  |
| --- |
| **<textarea** name="message" **rows="10" cols="30">** The cat was playing in the garden. **</textarea>** |

The **rows** attribute specifies the **visible** number of lines in a text area.

The **cols** attribute specifies the **visible** width of a text area.

Once limit of rows or cols crossed, we are going to get “scrollbar” like below. We have vertical scrollbar and horizontal scrollbar.



**The <button> Element:**

|  |
| --- |
| <**button** type="button" **onclick="alert('Hi..AZAD...')">**Click Me!</button> |

Here “**onclick**” **attribute** is called as **javascript event**, we are having so many JS events, all are mostly started with “**on**XXXX” keyword.

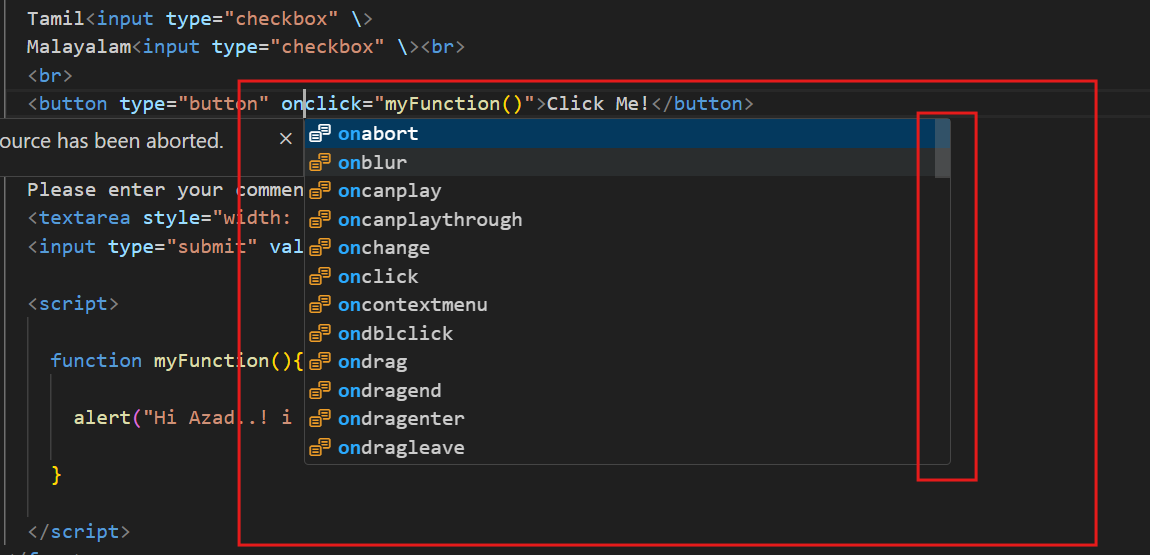
alert() is called as Javscript function. It is a predefined function. Means it is given by Javascript developers. We are not implemented this alert( )function.

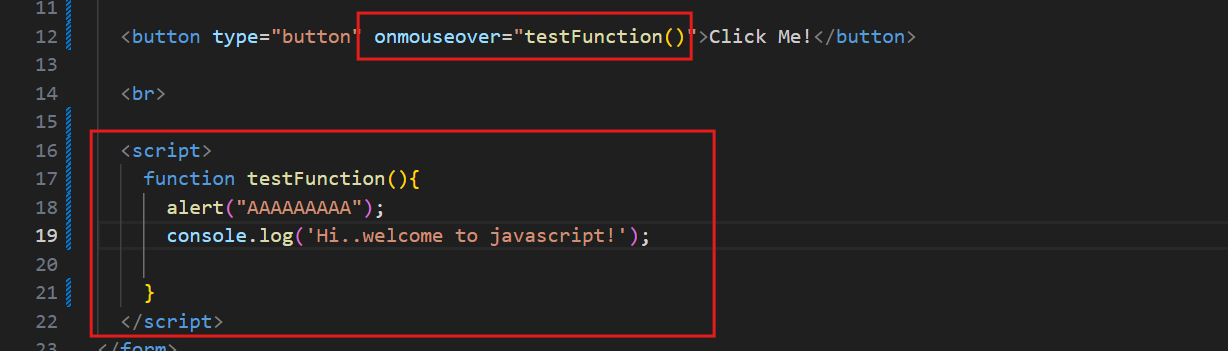
We can create custom function also; in future we are going to experience how to create custom functions.

**Note:** Always specify the type attribute for the button element. Different browsers may use different default types for the button element.

**For example below are some of JS events.**

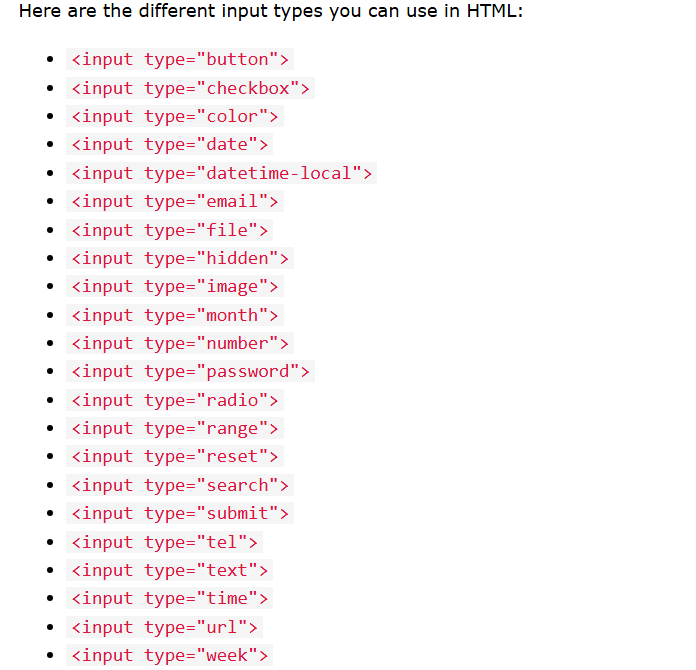
**onclick(), ondbclick(), onblur(), onchange(), onkeydown(), onkeypress(), onkeyup(), onfocus(), onload(), onmouseover(), onmouseout(), onsubmit().**







**HTML Input Types:**



**What is masking?**

The characters in a password field are masked.

Eg: \*\*\*\*\*\*\*\*4567 🡪 it is bank number of Aadhar number masked like this for **security purpose**.