# HTML

HTML stands for Hyper Text Markup Language

**Markup Language** means it includes metadata **for annotating** the document (web page) which is visually distinguishable from how user sees the document.

An analogy about the above definition:

When you open a browser and search something, you can see some text on tab heading and you will see some text on the webpage, we simply **markup** the language text with tags (viz, title and body tags)

Ex:

<title> title you want to give to the page</title>

<body>

Text that you want to put as body

</body>

Basically the text between tags is displayed differently based on the tags that are used around the text

Basic tags: head, title, body, (headers) h1 to h6, (paragraph) p.

A p tag is used so that a web page can be (conveyed using HTML) that the whole text in this tag is to be displayed as a whole and no empty lines should be used unless specified

## Elements in HTML page

### Lists

A common form factor on a page is lists

HTML has 2 main types of lists

* Ordered lists
  + Set with <ol> </ol> tags
  + Is numbered
* Unordered lists
  + Set with <ul></ul> tags
  + Has bulleted points

Items in either ordered or unordered lists will have an <li></li> tags

We can nest one list in another

### Div

* Can be created using <div></div>
* This will allow us to separate HTML in to different sections, so that we can apply different styles (CSS) to different division sections
* Is a **block level** (i.e, styles applied to a div will apply to all elements in it)

#### Span

* Similar to div, but is an **inline container** used to markup a part of a text or a part of a document i.e., using this we can apply css styling to only to the elements that we want to style
* Used with <span></span> tags

##### Example for Difference between div and span

<div style="color: green;">

    <p>this si para1</p>

    <p>this is para2</p>

</div>

<p>this is <span style="color: red;">para3 </span> </p>

### Tags with mandatory attributes

In general HTML supports many tags which does not need attributes mandatorily, but there are 2 important tags which need attributes mandatorily

#### Image tag

This takes src as an attribute where we pass the path of the image where it is stored

Ex: <img src = “path of the image” alt = “alternate text that you want to display if the path of the image is not correct or if the image is broken”>

#### Anchor tag

This tag takes the URL which stores the information about the link to which you want the user to get redirected to, by clicking on it

Ex: <a href= “URL”> text that you want to display as place holder for the URL </a>

### Table

To represent tabular data, we need to have table tag, in the below example the hierarchy is as follows,

Table is divided in to thead (optional) and tr, thead is further divided in to th each th represents one column name, each tr is further divided in to td, which represents each cell of data

Border is an **optional** column

Ex:

<table border="1">

    <thead>

        <th>col 1</th>

        <th>col 2</th>

        <th>col 3</th>

    </thead>

    <tr>

        <td> data 1</td>

        <td></td>

        <td>data 3</td>

    </tr>

</table>

### Forms

One of the most important elements in HTML

It is what makes the web page interactive, a form simply takes input from the user (this can be done in more than one form)

Ex: user entering search terms on google page is handled through forms

Inside an HTML form we have **input** and **label** attributes, based on the attribute type (input or label) the nature of its display and functionality changes

Label – what is displayed on screen

Input – what user inputs, this is sent to server for further process

In the below example you have used a parameter name, this is useful when you are trying to view the payload

#### Ex1:

<form action="two.html">

    <label for="fname"> First name:</label>

    <input id="fname" type="text" name="fname" placeholder="enter your email here..."><br>

    <label for="lname">Last name:</label>

    <input type="text" id="lname" name="lname"> <br>

    <input type="submit">

</form>

Explanation for terms in above example:

Action – defines what the form should do (should it route to another html page or anything else specific as per data provided to this attribute)

Id in input tag: this is used to uniquely identify and relate, for example- you can tag “First name:” label to the input tag with id “fname”

type in input tag: it accepts different parameters that give different functionality, email will only accepts text with @ and ‘.’ In it, password will display entered text as \*

name – this attribute is used so that the data collected in input fields can be navigated (along with the URL) i.e., if you use the above code and type some data for both the input fields and type submit, then you will see that the data that you typed will also be sent along with the URL mentioned in action parameter

#### Ex2:

<form action="two.html">

        <p>Are you from the inside or outside of US?</p>

        <label for="in">Inside:</label>

        <input type="radio" id="in" name="loc" value="in">

        <br>

        <label for="out">Outside:</label>

        <input type="radio" name="out" id="loc" value="out">

        <br><br>

        <input type="submit" value="send">

    </form>

Explanation for above example:

You can see same name being used for both the radio buttons above. This is because, in order to only select one of the radio buttons available, you need to share the same name, also the value is different for both the radio buttons, because to distinguish which radio button you clicked while clicking on submit button.

#### Ex3:

<form action="two.html">

        <h3>please rate your stay</h3>

        <select name="rating" id="rating">

            <option value="great">3</option>

            <option value="OK">2</option>

            <option value="Bad">1</option>

        </select>

        <br>

        <label for="comments">please enter comments:</label>

        <br>

        <textarea name="comments" id="comments" cols="30" rows="10">

        </textarea>

        <br>

        <input type="submit" name="" id="">

    </form>

Explanation for above example:

To have dropdown we use select label, to have options in the dropdown we use options label

To have a few sentences as input we use label textarea

## Relation between HTML and HTTP

* When you visit a website, you may have noticed the URL starts with HTTP.
* HTTP stands for Hypertext Transfer Protocol and is the protocol used to enable communication (HTML Transfer) between a client and server.
* HTTPS just means that the HTTP is secured through encryption.
* There are many methods available using HTTP, such as:
  + GET
    - Requests data from a source
  + POST
    - Sends data to a server

## Global attributes

Attributes that can be used in any tag are called global attributes, ex: id, class

# CSS

Need of CSS: even though we have style attribute as an inline attribute, imagine a html document where you will be having 100 paragraph items and you need to add same style to them. It is almost a tedious task. So to avoid this we can have a css file (which will be linked using link attribute in head part of the current HTML file) or as a CSS block in the HTML file. In the below code you can see 2 link tags, they are sequential and will load one after the other.

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet"

  href="https://fonts.googleapis.com/css?family=Moo+Lah+Lah">

    <link rel="stylesheet" href="master.css">

</head>

<body>

    <div class="divClass">

        inside div 1

    </div>

    <p> text od para</p>

    <h1>heading 1</h1>

</body>

</html>

h1{

    color: rgb(64, 226, 43);

}

p{

    color: red;

    background-color: blue;

}

The above code will have the red color text and blue background color for all the para elements where the current css file is linked in the head of the HTML document

## Specificity in CSS

Specificity priority means the order of priority in which the elements are to be styled

The smaller elements are given higher priority

Ex:

If you have a span tag which is inside a div tag and the div tag is obviously inside the body tag

Order of specificity priority is

Span>div>body

The span styling overrides the div tag priority and body priority

## Use of id, class in CSS styling

When the document becomes large, it might not be possible to apply same styling to all elements.

Analogy: if there are multiple paras, as different paras need different styling

if a few paras need the same styling, declaring **same class for them** and then perform styling in CSS

if only a particular para needs a particular styling, then declaring an **id** for that para will do the work

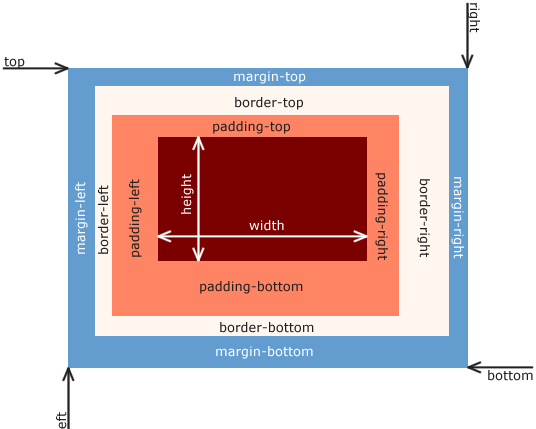
in styling you have to use

# before an id (similar to hashtags in Instagram where you have user **id**)

**.** before a class (similar to OOPS paradigm in python where you use dot **.**  to call a method of **class**)

## Box model

The maroon color (inner most) box is the content



An analogy of padding, border, margin: Suppose you have a content and you drew a border. So within the border if you want to let the content stick to the bottom of the border, you need to give padding at the top, if you want to have a space to the left of the border you have to use margin-left

# Bootstrap

You can think of bootstrap as a combination of a CSS file which is already created for you and an HTML file which has ready to use arranged HTML components

It also uses a **grid system** to easily resize your webpage for mobile or tablet or desktop screens.

Website: <https://getbootstrap.com/>

This website has examples, documentations, Icons and few other things

You can look at example templates like <https://getbootstrap.com/docs/5.2/examples/pricing/> in <https://getbootstrap.com/docs/5.2/examples/> where you can copy, try, play and experiment and learn

And look at documentation for components that you want to include.

What happens on different screen sizes when you are using the bootstrap framework?

Entire screen is divided in to 12 columns

A picture containing background pattern

Description automatically generated

Based on the item (count) you can divide them in a way (if possible) such that you will get 12 as the multiplied value

* The grid system call will make use of the class=”row”
* Inside of a row class, we then have the following format:
  + col-ScreenSize-NumberOfColumns
    - ex: col-md-6
* So we can define how the columns should be shown when the screen gets resized.

## Misc

* You can stack one class on another if you want to get the functionality of both the classes
  + <button type="button" class="btn btn-primary">Primary</button>
    - Here you can see two classes btn, btn-primary
* For Django you need forms, buttons, navbars, grid system and layout, already available at https://getbootstrap.com/docs/5.2/getting-started/introduction/