html => hypertext markup language
=> data presentation
=> UI designing (sign up, login, registration, search form…)
= static webpages

CSS => Cascading Style Sheets
=> used to change look & feel of webpage (html elements)

JS => JavaScript
=> its back-end for html/css (Front-end)
=> it provides logical support or client validations

Html/css/JavaScript ➔ static web site designing

Html/css/JavaScript+angular or reactJS➔ front-end developer
UI/UX developer

advJava/spring/asp.net/python/php/nodejs➔back-end dev

front-end+back-end+DB+config+versioning➔full stack developer

full stack web dev:

html/css/JavaScript/anguler/SST/bootstrap/wordpress/xml-werservice

server side tech's:
node.js/servlet/asp.net/php/cgi-perl/py

Network: Collection of computers interlinked together is called network. First network name is **ARPANET** (Advanced Research Projects Agency Network). First protocol in IT industry is FTP (File Transfer Protocol).

Internet: Internet stands for international networking.
1990
The Internet is a network of connected computers. No company owns the Internet; it is a cooperative effort governed by a system of standards and rules. The purpose of connecting computers together, of course, is to share information.

Internet is a collection web application,
Web application is group of web pages
Web page is group components (means heading, para, image, button, tables, …)

A Brief History of the Web

The Web was born in a particle physics laboratory (CERN) in Geneva, Switzerland in 1989. There a computer specialist named **Tim Berners-Lee** first proposed a system of information management that used a "hypertext" process to link related documents over a network. He and his partner, **RobertCailliau**, created a prototype and released it for review. For the first several years, web pages were text-only. It's difficult to believe that in 1992, the world had only about 50 web servers.

TimBerners-Lee➔ internet (1989-1990)

- ⇨ Html (HyperText Markup Lang)
- ⇨ http (hyper Text Transfer Protocol)
- ⇨ W3C org

The World Wide Web Consortium

World Wide Web Consortium (called W3C) is the organization that oversees the development of web technologies. The group was founded in 1994 by TimBerners-Lee, the inventor of the Web, at the Massachusetts Institute of Technology (MIT).
Tim Berners-Lee (WWW/HTTP), Cerf & Kahn (TCP/IP), Baran, Davies, Kleinrock & Roberts (packet networking), Bob Metcalfe (Ethernet).

WHAT IS APPLICATION OR SOFTWARE?

Automation of manual business operations by using a programming language.

TYPES OF APPLICATIONS OR SOFTWARES

We can create an application or software in fallowing flavors:

1) **Desktop:** The applications which are installable in local systems are called desktop applications.
2) **Mobile:** The applications which are installable in mobile phones or tablets downloaded from play store for android and apple store for ios.
3) **Web:** The applications which are deployable in any server and can be accessible from any location using browser.

WHAT IS WEB APPLICATION?

Web applications are network enable applications. We can deploy any web applications in servers and we can access them over network using server ip address and application name.

In computing, a **web application** or **web app** isaclient–server software **application** which the client (or user interface) runs in a **web browser** and it contains web documents in the form electronic pages(web pages).

A web application typically contains fallowing three layers:

Presentation layer is a user interface (views) which are accessible from any web browser.

Business layeris a server-side program which is nothing but automation of business rules. Client layer will interact with business layer to persist data.

Data layeris database software where we can store client related data. Business layer will interact with data layer.

How the Web Works

1. When you connect to the web, you do so via an Internet ServiceProvider (ISP). You type adomain name or web addressinto your browser to visit a site;for example: google.com, oracle.com, microsoft.com.
2. Your computer contacts anetwork of servers calledDomain Name System (DNS)servers. These act like phonebooks; they tell your computerthe IP address associated withthe requested domain name.An IP address is a numberof up to 12 digits separatedby periods / full stops. Everydevice connected to the webhas a unique IP address; it islike the phone number for thatcomputer.
3. The unique number that theDNS server returns to yourcomputer allows your - browserto contact the web serverthat hosts the website yourequested. A web server is acomputer that is constantlyconnected to the web, and is setup specially to send web pagesto users.
4. The web server then sends thepage you requested back to yourweb browser.

What is web browser?

It is client-side lightweight software installed in client machine. It sends http request from client to server;ittakeshttp response from server.
Browser provides navigation among web pages, and browsers executes html, css, JavaScript files and displays output to user.

List of Computer Browsers:
Internet Explorer(1995), Opera(1995), Mozilla Firefox(1998), Safari(2005), Google Chrome(2008) etc…

List of Mobile Browsers:
Mobile Safari (iOS), Android Browser (Android), BlackBerry Browser (RIM), Nokia Browser (Symbian), Opera Mobile and Mini (installed on any device), Internet Explorer Mobile (Windows Phone), Silk (Kindle Fire) etc…

Server

A **server** is a computer or system that provides resources, data, services, or programs to other machines, known as clients, over a network/inet. In theory, whenever computers share resources with client machines, they are considered **servers.**

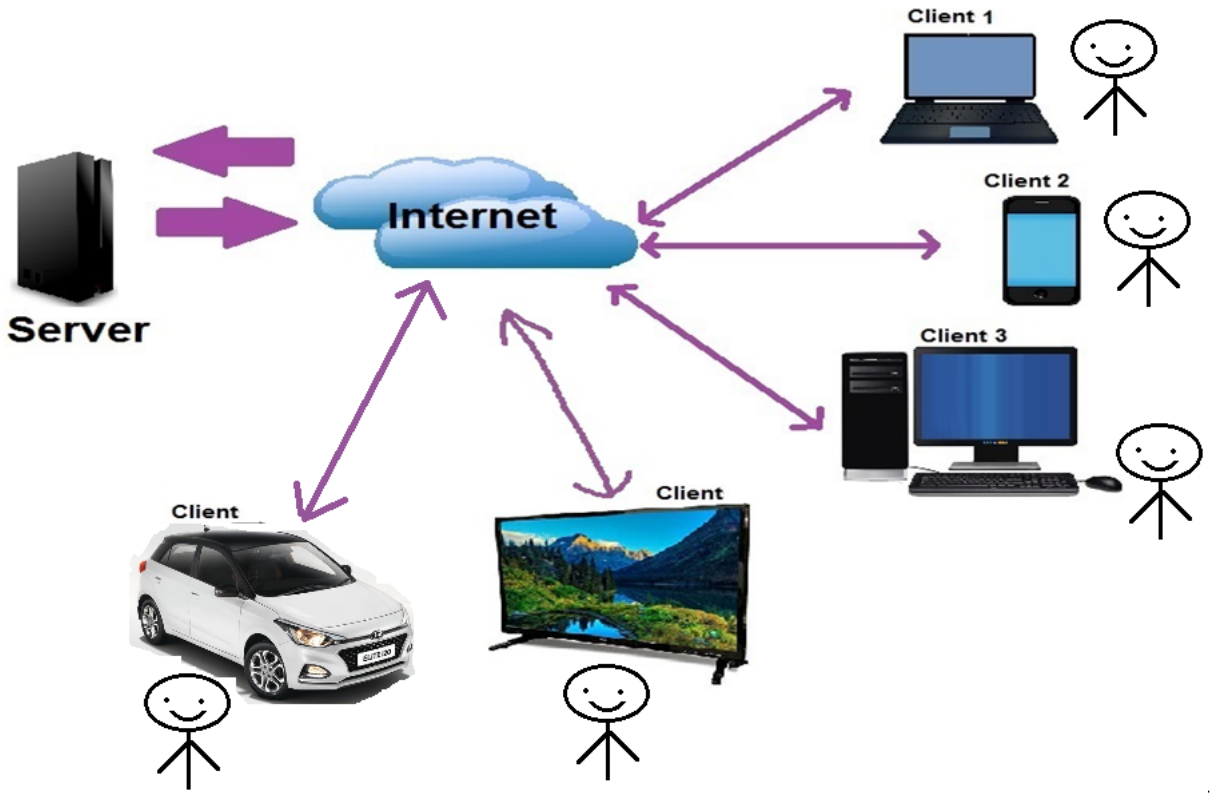
a **server** stores all the data associated with the websites that are hosted by it and shares that info with all computers and mobile devices (like yours) that need to access them.

Client

A client is a electronic device that connects to and uses the resources of a remote computer, or server.
Client maybe a desktop or a laptop or a tablet or a mobile phone or a TV etc.
The device which is used by the user is called as “Client”.

User

The person who is working on/operating client machine is known as User or end-user.



Client:

It is a machine or device (desktop or laptop or tablet or mobile phone or TV etc), which can access the data from server machine.
The device which is used by the user is called as "Client", person who is working on client machine is known asUser.

Email: Electronic mail services. It is a free service to communicate with other internet users. Email is invented by **Shabeer Bhatia.** Sabeer Bhatia is an Indian entrepreneur who founded the webmail company Hotmail.com.
SMTP: Simple Mail Transfer Protocol. It takes care of delivering emails from one server to another.
MIME: Multipurpose Internet Mail Extensions. It exchanges different kinds of data.
Blog: It is daily updating website or webpage. Every post displayed in reverse chronological order.
Forum: It is an online discussion website to exchange resources each other.
Http: It is a transfer protocol to exchange hypertext documents in the world wide web.
Http(s): Secured transfer protocol to exchange hypertext documents with the help of SSL(ciphertext).
Ciphertext is encrypted text. Plaintext is what you have before encryption, and **ciphertext** is the encrypted result. The term cipher is sometimes used as a

synonym for **ciphertext**, but it more properly **means** the method of encryption rather than the result.

HOW MANY TYPES OF WEB APPLICATIONS WE HAVE?

A webpage is an electronic page developed on HTML. It is classified into two types.

Static webpage: A user unable to interact directly with these webpages. Eg: HTML, CSS

Dynamic webpage: End-user can able to interact directly with these webpages. Eg: HTML, CSS &Javascript

Collection of webpages or web documents are called web application(website). These are classified into two types:

STATIC WEB APPS: The applications which can't able to handle business logic are known as static web apps.Static apps will contain only client layer.We can develop static web applications using HTML. To provide look and feel to these static pages we can use CSS. To handle client layer business logic we ca use Javascript.We can't able to maintain end user interaction(state) using static web apps.

DYNAMIC WEB APPS: The applications which can able to handle business logic are known as dynamic web apps.These type of apps contains at least 2 layers client and business. If we need to store client data then these application contains data layer too.We can develop client layer using HTML, CSS &javascript and business layer using any one of the server programming language like .NET, JAVA/J2EE & PHP etc...We can store end user data using any database like mongo db, MS-SQL, MySql, Oracle etc.

What is HTML?

It is specially designed hypertext for web browsers, with meaningful tags or elements in simple English language.

HTML Versions

From W3C organization there are fallowing versions released.

Version	Specification	Release Date
1.0	**N/A (HTML 1.0)**	**1-Jan-1994**
2.0	**HTML 2.0**	**24-Nov-1995**
3.2	**W3C: HTML 3.2**	**14-Jan-1997**
4.0	**W3C: HTML 4.0**	**24-Apr-1998**
4.1	**W3C: HTML 4.1**	**24-Dec-1999**
5.0	**WHATWG (Adv Markup Language For Mobiles)**	**28-Oct-2014**
5.1	**W3C: HTML 5.1 (Adv Markup Language For Small Electronic Devices)**	**-Nov-2016**
5.2	**W3C: HTML 5.2**	**14-Dec-2017**

HTML Intro

1. HTML wasdeveloped by “Tim-Berners-Lee”, released in 1994 and maintained by W3C Org.

2. HTML stands for “Hypertext Markup Language”.

3. Hypertext” means the text that can be transferred from internet server to internet client.

   "Markup Language" means which syntax will be in the form of tags or you simply "markup" a text document with tags that tell a Web browser how to structure it to display.

4. Technically, HTML is not a programming language, but rather a markup language.

5. HTML is used to design "**static web pages**", means HTML is used to create elements (such as headings, paragraphs, icons, menus, logos, images, textboxes, button etc) in the web pages.

   static webpage means, that pages always showing same information.

6. HTML is very easy to understand (no pre-requisites).

7. HTML is “client side tech”. That means the html code executes on the client browser but not in server.

   web tech:

which sw are supporting to design web pages or providing API to dev web coding those sw are called as web tech.

>**client side tech** ex: html/css, js, jquery, BS ...

used for static web pages.

bw rec source code & trans after execution then produced the output.

>**server side tec**h ex: servlet, jsp, asp.net, php, cgi, nodejs, cold fusion ...

dynamic web pages.

code trans, execute with in server only, and produced output, this output sent to client machine.

8. HTML is supported by all the browsers such as Google Chrome, Mozilla Firefox, Microsoft Internet Explorer, Safari, Opera and other browsers.

9. HTML is used in all real time web sites today; html is the only language available in world for designing Webpages.

1.The file extension either "filename.html" or "filename.htm"

2.HTML is an interpreter-based language. That means the HTML code will be converted into machine language in +. Browser interprets HTML code.

Translators:converting high level code (human) into machine level code (MP/OS) is called as translation. who performs this operation those called as translators.

types: >compiler ex: c, cpp,...

>interpreter ex: html, js, oracle,...

>assembler

3.for working html no need installs any software, and **browse**r is responsible for executing & producing output of html programs.

4.html is 100% error free programming.

5.HTML is not a case sensitive language that means you can write the html code in either upper case or lower case.

how design& execute html programs

- open any text editor (sw) and type program.

notepad, editplus, notepad++, textpad, sublime, ms vs, word, atom, coffee, ...

- save that program with any name (.html or .htm) and anywhere in system.

but filename must be single word (space is not allowed).

> execution:

1st Approch:goto file location, then dbl click on file

2nd Approch:goto file location, then right click on file and click on open then select browser

3rd Approch:open any browser, then goto address bar and type filename with address.

d:\siva\test.html e:\test.html

Tag:

- A tag is a keyword, enclosed within "<" and ">" in HTML language.
- It is special kind of text placed between left angular brace and right angular brace(<tag_name>).
- Tag is predefined program, program is instructions / command to browser.
- Tag is used to display some specific output in the web page.
- browser was not identified the tag; it shows blank page or it prints as text.
- tags also represented as elements.
- tag has some attributes(properties), those are used to change look & feel (components or output).

types of tags:

in html we have **two** types tags, those are:

>**paired tags**

contains open tag and closing tag.

opening tag specifies starting point of operation/output, closing tag specifies ending point of operation/output.

Syn: **<tagname>**something**</tagname>**

ex: <html> ... </html>
<head> ... </head>
<body> … </body>
<script> ... </script>
<style> ... </style>
 …

note: paired tags also called as body-full tags

>**unpaired tags**

contains only open tag.

VOID => ITS not RETURNING ANY VALUE

Syn: <tagname> or <tagname/>

ex:
 <img/><input/>
<hr>
<link>

note: Unpaired tags also called as body-less tags

Structure of HTML

as per **W3C** we have to follow the following structure to design web pages (but it's not comp).

<!DOCTYPE html>
<html>🡐web page/document designing starts here
<head>
-> non-content sec(non-result)
-> settings/internal info about page
-> 1st executed sec
Ex: title tag, link tag, style tag, script tag, meta tag
</head>
<body>
-> content sec(result)
->it contains page designing
-> 2nd executed sec
Ex: form, h1, h2, h3, h4, h5, h6, p, div, table, img, a, button, audio, video, iframe, etc…
</body>
</html>🡐web page/document designing ends here

generally, html page contains three parts, those are:
>versioning section
>head section
>body section

versioning section

this is providing information to browser which version we are using in webpage/program. So, browser is interpreting code and producing output as per given specification.

Syn:
<!DOCTYPE htmlversion-url>

HTML4.0:
<!DOCTYPE html public "-//W3C//DTD HTML 4.0//EN"
"http://www.w3c.org/TR/html4/strict.dtd">

XHTML:
<!DOCTYPE html public "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3c.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

Html+xml =>xhtml

HTML5:

<!DOCTYPE html> ← **current version**

strcit.dtd file (document type definition), it contains definitions of tags, specifications.

doctype is not case-sen, so we type in any case.

ex: DOCTYPE => valid (recommended)

doctype => valid

DocType => valid

html tag

the <html> tag represents starting and ending of html program. html tag contains two child/sub tags those are head tag and body tag.

head tag

head tag represents non-content section (means not output) of the web page.
this information doesn't appear on webpage/in browser (it's called as non-content), but it's used internally by the browser.
this tag is used to set icons, title, to provide some meta data (info about web app), css settings, java scripting etc...
head tag contains some child/sub tags, those are

Syn:

```
<head>
<link>
<title></title>
<meta>
<style></style>
<script></script>
   ...
</head>
```

body tag

body tag represents content information (means output) of the web page.
this information appears on webpage/in browser (it's called as content).
this tag is used to design UI or to display output.
body tag contains so many child/sub tags.
some of tags:

<body>

```
        <form>
        <h1>
        <h2>
        <p>
        <div>
        <input>
        <a>
        <audio>
        <video>
        <iframe> etc...
    </body>
```

html is collection of tags(elements) and attributes.

comment lines

comment lines are to provide some description about of our program.

Syn:

```
<!-- comments -->
```

comments are not executed by browser.

heading tags

these tags are used to print data/text in heading format.
html providing 6 heading tags, those are h1, h2, h3, h4, h5, h6.
These 6 tags are used to display headings in different sizes.
six tags are paired tags and block level elements.

Syn:

```
<h1> text </h1>
<h2> text </h2>
<h3> text </h3>
<h4> text </h4>
<h5> text </h5>
<h6> text </h6>
```

Note: inside body section we can repeat any tag and no.of times.

p tag

> p stands for paragraph.
> this tag is used to display/print more lines of text (paragraph)
> its paired tag and block level.
> browser display an empty line(gap) between paragraphs
Syn:

** text or info **

Note:

>browser/html doesn't accept more than one space (space bar & tab key), means while designing of program we given more space but browser prints only one space.
>browser/html doesn't accepts enter key (line breaking), means while designing of program we use enter key but browser prints data without breaking line.

br tag

- br stands for break line (enter key)
- its un-paired

**Syn:
 or
**

Html entities

Special characters or html operators

Syn: &entity;

** © ™ ® € £**
¥ < > ¼ ½ ¾ etc…

Html hexa-decimal operators, these operators are starts with #
Hexa-dec base 16 ➔ 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
Syn: &#hexa-code;

Label tag

>label tag used for displaying prompting text.
>its paired tag, inline tag
Syn: <label> text </label>

Span tag

>span tag used for small textual data, like as error message, mandatory specification.
> in continuity of text, if we want to highlight couple of word or **letters**, we use span tag
>its paired tag, inline tag
Syn: ** text **

pre tag

> pre stands for pre-formatting (alignment)
> pre tag is used to print data/text, how we typed in same format
> pre is paired tag, block level
Syn:
<pre> text </pre>

formatting tags

<b>	**<u>**	**<i>**	**<strike>**	**<sub>**	**<sup>**
Ex: **Apple**	apple	*4568*	~~apple~~	H_20	a^2

All are paired tags
Inline tags

b tag or strong

> b stands for bold
> b tag used to print text in bold format
>both are paired tags & inline tags
Syn:

 text
 text

u tag

> u stands for underline
> u tag used to print text with underline (draws a line base of text)
> u is paired tag

Syn:

<u> text </u>

strikeout tag

> strikeout tag used to print text with line (draws a line middle of text)
>strikeout is paired tag

Syn:

<strike> text </strike>

superscript tag

>this tag used to display text top of upper line
> superscript is paired tag

Syn:

^{text}

subscript tag

>this tag used to display text bottom of baseline
> subscript is paired tag

Syn:

_{text}

I or **em** tag

>i stand for italic (inclined)
>i tag used to print text with little banding
>i is paired

Syn:

 text
 text

All 6 tags are paired tags &inline tags

title tag

this tag used to set the title for a webpage, means every webpage

they have individual title.
its paired tag.
<title> is the sub tag of <head> tag.
Web site => 10 web pages =>Title 10 times

Syn:

<title>text**</title>**

Note: one web page/one title

Link tag

Link tag used to set the icon/logo for a webpage.
Un-paired tag.
<link> is the sub tag of <head> tag.

Syn: <link rel="stylesheet" href="filename"/>

Relative
Hyper reference =>.jpg .bmp **.png** **.jfif** .gif .tif **.ico**

Preferable image size:

18px X 18px
20X20px
30X30px
40X40px

Attributes

> attribute is a special feature/setting/property of a tag.
> attributes are used to change the default look/feel of data(elements).
> every tag they have attributes

Syn:

<tagname attribute="parameter" attribute='parameter' ...>⬅ Html

Width="100px" height=200px

Note:

- parameter means the value of attribute.
- Parameter should enclosed within " " or ' ' or without quotes.
- Every attribute must be separated by a space

types:

as per html4 we have 3types of attributes, those are

>**global attributes** (core)

Ex: id, name, class, style, align, width, height, title etc…

>**specific attributes** (personal)

Ex: rel, href, src, colspan, rowspan, action, alt etc…

>**event attributes** (dynamic)

Ex: onclick, onload, onfocus, onblur, onchange, onsubmit, onkeypress, Oncopy, onpaste, oncut, onchange, ondblclick, oncontextmenu, onmousemove, onmouseover, onmouseexit etc…

>**optional attribute** <== html5

Ex: lang, type, method etc…

global attributes:

these attributes are common for most of tags (99% of tags)
those attributes are:

class, id, name, style, align etc...

ex:

```
<h1 class="" id="" name="a" style="" ...>
<p id="" class="" name="b" style="" ...>
<pre class="" id="" name="c" style="" ...>
```

specific attributes:

these attributes are specific to some tags/elements only (not common).
those attributes are:

src, href, rel, target, colspan, rowspan, alt, placeholder, poster, loop, etc...

ex:

```
<a href="url" ... >
<imgsrc="" ...>
<audio controls>
```

event attributes:

these attributes are used to some logical operations.
logical operations we can perform by using JavaScript, these also called dynamic attribute.
attributes are:

onclick, oninput, onfocus, onexit, onload, onchange,onblur, ...

ex:

```
<button onclick="js code/fun1">
<body onload="js code">
```

optional attributes:

these attributes are not comp to specify/to use.
these type attributes are supported since html5.0.
those attributes are:

lang, type, method ...

ex:

```
<style type="text/css" ...>
<script type="text/javascirpt" ...>
<link type="image/jpg" ...>
<form method="get" ...>
```

categories of attributes:

html attributes ➔<tag attribute="value">
css attributes ➔<tag style="css-attribute:value; css-attribute:value;...">

CS S➔ change look/feel of html elements

Css provide only styles but not tags

Style is group of attribute/properties

Different ways to implement css:

1st Approch (inline):

Html tag and css properties are defined With n the same line

<tag Style="attribute:value; attribute:value; …">

2nd Approch (internal):

Html tags and css styles are designed in the same program, but not in same line.

Internal css should be implements in Style tag, style tag must be sub tag head tag.

```
<style>
h1{
        attribute:value;
        attribute:value;
        …
  }
  Selector{
        attribute:value;
```

```
            attribute:value;
            …
        }
        …….
    </style>
```

3rdApproch (external)
Css styles are designed in separate file and should be save with ".css", and html code saved with ".html"
Use link tag for mapping css file to html file
Syn: **<link rel="stylesheet" href="filename.css"/>**

note:

- css attributes we can't use in place of html attributes.
- html attributes we can't in place of css attributes.
- "Id" attribute we are using while writing javascript code.
- "Name" attribute we are using while writing server-side code. (servlet, jsp, asp, php, noedjs)

images

> "img" tag is used to display images on webpage.
> in one webpages we can display any no.of images and any type of images.
> it is strongly recommended to place all images in side root folder or create sub folder with name images in root folder
>its un-paired tag, and its inline element
Syn:

```
<img attributes/>
```

.jfif .svg.jpg .bmp .gif .tif .png .webp

attributes:
src => to specify which img you want to display
width => width of image (pixel)
height => height of image (pixel)
title => it is used to specify tool tip. (whenever mouse pointer comes on top of image)
alt => alternative text, if image not loaded in webpage/not display, we want to display text message to user it called as alt

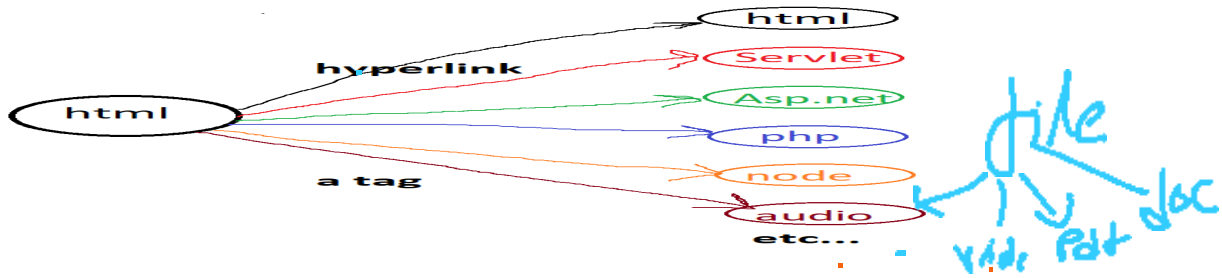
+

global attributes

opacity: 0.5;
filter: blur(5px);
brightness(125%)
contrast(135%)
grayscale(100%)
invert(100%)
hue-rotate(180deg)
saturate(8)
sepia(100%)
drop-shadow(8px 8px 10px green)

hyperlinks

> a stand for "anchor"
>"a" tag is used to create hyperlink, hyperlinks are used to move from one webpage to another webpage.
>whenever user clicks on the hyperlink, it moves to the specified page.
> destination page sometime within same application or other application.



>web application basically contains links to other pages, so it's very commonly used tag.
> by default, every browser provides built-in style for each hyperlink,
i.e blue color+handsymbol+under line.
we can customize these styles by using CSS.
> its paired tag, and inline element

Syn:

<a attributes>Display Text
<a attributes> <img>

attributes:

href : hyper reference, used to specify the address of webpage or web site,
i.e whenever user clicks on this link, which page you want to open

url may be html page, server-side file, image, audio file, video, pdf file, documents etc...

href="url"
"https://www.abc.com/login.aspx"
"" ➔self-calling
"." ➔ home page of web site/home dir of web application
"#id" ➔ it creates book marks (moving within same page)

target : where you want open destination page
_blank ==> opens the link in a window/tab
_self ==> opens the link in current working tab/window (its default)
_parent ==> opens the link in parent frame
_top ==> opens the link in full body of window
framename ==> opens the link in specified frame

html colors

html supports 3types of patterns, those are
> named colors
> RGB colors
> Hexadecimal colors

named colors:
>it supports to write direct colorname
>we have some limited colors
ex: white, black, red, green etc...
> color names are not case-sen

RGB colors:
>RGB model specifies that the composition of 3 basic colors (Red, Green, Blue)
>RGB produces 16millions colors.

Syn: **rgb**(red,green,blue)
red => 0 - 255
green => 0 - 255
blue => 0 – 255
ex: **rgb**(10, 45, 201) 401%255 ➔146

Hexadecimal number colors:

>Hexadecimal model is the shortcut for rgb model
>Hexadecimal system ranges from 0 - 15
0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
Syn: #RRGGBB 1,2 red 3,4 green 5,6 blue
ex: #1a4b68
#RGB
ex: #3d7

Note: in realtime "Hexadecimal model" is recommended.

these colors we can use for foreground color, background color, border color etc..
for setting colors we have some attributes, those are
color ➔ to set/to change foreground color (text color)
background-color ➔ to set/to change background color
border-color ➔ to set/to change border color (line color)
box-shadow ➔ to set/to change shadow color
text-shadow ➔ to set/to change text shadow color
Note: all these are CSS attributes.Support by Most of html tags

Gradient colors

background: #FC466B; /* fallback for old browsers */
background: -webkit-linear-gradient(to bottom, #3F5EFB, #FC466B); ➔ Chrome 10-25, Safari 5.1-6
background: **linear-gradient**(to bottom, #3F5EFB, #FC466B); ➔W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+

linear-gradient(direction, color1,color2,…color-n)
dir: to left (r=>l)
to right (l=>r)
to top (b=>t)
to bottom (t=>b)

background: **linear-gradient**(to bottom, #3F5EFB 40%, #FC466B 60%);

-webkit-linear-gradient(to left, #3F5EFB, #FC466B);

linear-gradient(to left, #3F5EFB, #FC466B);

background: radial-gradient(circle, rgba(2,0,36,1) 0%, rgba(38,38,162,1) 60%, rgba(0,212,255,1) 100%);

radial-gradient(shape, color1, color2, …color-n)

radial-gradient(circle, rgb(131,58,180) 0%, rgb(29,166,65) 50%, rgb(252,176,69) 100%);

radial-gradient(circle, rgba(166,29,142,1) 57%, rgba(100,180,111,1) 78%, rgba(69,252,96,1) 100%);

Note: while applying gradient colors we have to use "**background**" property in place of "**background-color**".

working with list tags

these tags are used to display data/info in points wise.
html supports three types of list, those are

Ordered list➔ numbering
Unorderedlist ➔ bulleting

ol tag

>ol stands for "Ordered List".
>it is used to display the text(names, colors, team names, course name...) with numbering.
>it supports 5types numbering, those are **1, A, a, i, I**. by default it displaying in number.
>by using "ol" tag we can create ordered list
>ol is paired tag & block level element

li tag

> li stands for "list item"
> li is sub tag of ol tag
> li tag is used to print text/data in points wise
> li is paired tag & block level element
Syn:

<ol attributes>

```
        <li> text </li>
        <li> text </li>
        <li> text </li>
    ...
</ol>
```

ol attributes:
type : which type numbering to display (Default is 1)
start : from where u want to start numbering (default is 1)
reversed : to displaying numbers in desc order

li attributes:
value : used for restarting numbering with specified value

ul tag

>ul stands for "Un-Ordered List".
>it is used to display the list of items(names, colors, team names, course name...) with bulleting.
>it supports 3types bulleting, those are **dot, circle, square**. by default, is dot.
>by using "ul" tag we can create un-ordered list items
> ul is paired tag
>"li" tag used for creating list items
Syn:

```
<ul type="dot/circle/square">
<li> text </li>
<li> text </li>
<li> text </li>
  ...
</ul>
```

dl tag

>dl stands for Definition list (since html5 description list)
>dl tag used for to display definitions/full forms (collection of definitions)
>its paired tag
> "dt" and "dd" are sub tags of "dl" tag
> "dt" stands for definition title, "dd" stands for definition data.
> dt & dd are paired
Syn:

```
<dl>
<dt>title/word</dt>
<dd>information</dd>
<dt>title/word</dt>
<dd>information</dd>
<dt>title/word</dt>
<dd>information</dd>
  ...
</dl>
```

fieldset tag

> this tag used for drawing a line/border around elements/tags.
> its paired tag and block level
> we can draw any no.of borders

Syn:<fieldset attributes>
<legend>text</legend>
Sub elements
</fieldtset>

attributes:

align : align of elements, it supports 3 alignments center, left, right
left is default align
border : style of line, thickness of line, color of line
width : width of box (size in %)

legend tag

>legend tag used for set title/heading for fieldset
>legend is sub tag of fieldset tag
>its paired tag

Syn:<legend attributes>Heading</legend>

attributes:

align :align of elements, it supports 3 alignments center, left, right
left is default align
color :

div tag

>div is a container, means its grouping elements/controls/components of html.

>inside div tag we can place any content like normal text or images.
>div tag is used to divide web page as no.of subpages/parts, each part is rep as div.
>for better maintained, effective design of web page and simplifying css code.
>its paired tag, and block level element
Syn: <div attributes>
contents

>one webpage may contains any no.of div tags.

display:flex; <== it displaying all elements side-by-side row wise or column wise
flex-wrap:wrap; <== it align element to next line
flex-direction <== it used to specifiy direction (order) of flex elements
flex-direction:row|row-reverse|column|cloumn-reverse;
flex-flow <== it combination of felx-wrap & flex-direction attributes.
flex-flow: direction wrap;

display:grid; <== it displaying all elements in rowsXcols
grid-template-columns <== no.of columns to display (width of columns)
grid-template-columns:col1 col2 col3....;

:autoautoautoauto; <== 4columns

:300px 400px 250px; <== 3columns

:30% 30%; <== 2columns

:30% auto 400px;
grid-column-gap: Npx; <== it provides a gap between column to column
grid-row-gap: Npx; <== it provides a gap between row to row
grid-gap:Xpx; <== it provides a gap between row-row & col-col with same size

Note: its applicable on nested tags, means outer tag only we can apply grid

table tag

>table tag is used to display the data in form rows & cols in the web page.
> a table is a collection of rows, each row is collection of cells/col/field.
> a table is represented as <table> tag, a row represented as <tr> tag, a colheading is represented as <th> tag, data rep as <td> tag.
> table heading is represented as <caption> tag.
><thead> tag is rep of table head part, <tbody> tag is rep of table bodypart and <tfoot> tag Is rep of table footer part.

table➔ it just comb rows & cols
caption➔ main heading of table
tr➔ table row, used to draw a row
th➔ table heading (col heading)
td➔ table data (col data)
thead➔ table head section
tbody➔ table body section
tfoot➔ table footer section

> all these 8tags are paired tags
table, tr, caption, thead, tbody & tfoot are block level tags
th & td are inline tags

Syn:

```
<table>
    <tr>
        <th>heading</th> <th>heading</th>
    </tr>
    <tr>
        <td>data</td> <td>data</td>
    </tr>
  ...
</table>
```

NOte:
<th> and <td> are sub tags of <tr>
<tr> is sub tag of <table>

table attributes:
border : border of table (0 means no border, 1-n border req)
align : alignment of table
width : width of table (%)

...

th& td attributes:

colspan : specifies the no.of columns to merge/expend

rowspan : specifies the no.of rows to merge/expend

...