**11/03**

**Web Applications**

**1. What is Network?**

A computer network comprises of group of computers connecting with each other for sharing information and resources.

ARPANET is the first computer network introduced in early 1960's by US-DOD.

[Advanced Research Project Agency Network]

Networks are classified into 3 major types

**a) LAN**

**b) MAN**

**c) WAN**

**2. What is Internet?**

- Internet is a "Wide Area Network" that connects computers all over the world.

- CERN [Council for European Research and Nuclear] labs developed the concept of Internet.

**3. What is Web?**

- Web is a portion of internet with restricted access.

- "Tim Berners Lee" introduced the concept of Web in early 1990's.

- Web uses the architecture of Request and Response.

- Web Standards are maintained by "W3C" World Wide Web Consortium.

**https://www.w3.org/**

**4. What are the principles of Web?**

a) Accessibility b) Internationalization c) Privacy d) Security

**5. What is the latest version of Web?**

A. Web3

**6. What web comprises of? What web contains?**

- Web Site

- Web Applications

- Blogs

- Micro Blogs

- Vlogs

- Podcasts

- Wiki

- Widgets etc.

**12/03**

**Web Application- Architecture**

**Backend**

**1. Data Center**

- Database

MongoDB

MySQL

**2. Application Center**

- Node JS

- Express JS

- API

**Frontend**

1. HTML

2. CSS

3. Bootstrap

4. Tailwind CSS

5. Sass

6. JavaScript

7. jQuery

8. React

**Tools**

1. GIT

2. Cloud

3. Testing [JEST]

**Front End Technologies**

**Setup your device for Web Technologies:**

1. Make sure that you have a device with Windows 8+ version

You can have other OS services like Linux, Mac.

2. Make sure that your device is having min 8+ GB RAM

3. Download and Install "Node JS" on your PC

- Node JS is used by developers to build web applications, servers, command line

tools and scripts.

- visit the official website of Node JS

[**https://nodejs.org/en**](https://nodejs.org/en)

- Download Installer package ".msi"

- Run the setup file to install

- Check the version from command prompt after installing.

C:\> node -v => make sure that you have the latest version of Node

4. Download and Install "Visual Studio Code" editor

- Editor is a software tool used by developers to build, debug, test and deploy

applications.

- There are various editors like sublime, webstrom, PyCharm, etc.

- Our editor is VS Code [Visual Studio Code]

**https://code.visualstudio.com/**

**13/03**

**Web Application Architecture**

- Backend

- Front End

**Setup Environment for Web Applications**

1. Install Node JS

2. Install Visual Studio Code [IDE - Integrated Development Environment]

**Setup VS Code for Web Designing:**

1. Open Visual Studio Code editor from your programs

2. Go to "Extensions" category

3. Search for following extensions and install on your PC

a) Live Server [ Ritwick Dey ]

b) Live Preview [ Microsoft ]

c) vscode-icons [vscode-team]

d) IntelliSense for CSS class names in HTML [Zignd]

4. Go to "Settings" of Visual Studio Code

5. Open Settings => Search Settings => Type "emmet"

6. Select the checkbox "Show Abbreviation Suggestions"

**Create a new Web Application:**

**FAQ's:**

**1. What is a website?**

A. Website is a virtual directory on web server.

It provides access to resources on server.

**2. What is Physical Path?**

A. Every website keeps its resources on server at a specific physical location, which

you can refer as Physical Path.

D:\erp-app => Physical Path

**3. What is Virtual Path?**

A. Website resources are accessed by using an URL [Uniform Resource Locator],

which is referred as Virtual Path.

http://127.0.0.1/erp-app

**4. What is Web Application?**

A. Application handles a business. It enables all business interactions.

It requires both client side and server side integrations.

**Setup Application:**

1. Open any PC location and create a new folder for your application

D:\fitness-app

2. Open Visual Studio Code

3. Go to File Menu => Open Folder => D:\Fitness-App

Note: Every application comprises of 3 environments

a) Development [Design]

b) Quality [test]

c) Production [deploy - go live]

4. Setup Development environment for your App

- Open Terminal in VS Code [Ctrl + `]

- Run the following command

*> npm init -y*

\* NPM is node package manager.

\* Package Manager is a tool required for installing, updating and uninstalling any library required for project.

\* There are various package managers like

a) Yarn

b) Bower

c) NuGet

d) Composer

e) Ruby Gems

f) NPM etc.

- It generates a new file into project by name "package.json"

\* It comprises of project meta data.

\* It is project configuration file.

- Add a new file into project by name "README.md"

\* It is a help document

\* Designed by developer for developer

- Add following folders into project

a) public

b) src

\* public is for static resources like images, docs, html etc.

\* src is for dynamic resources like css, scss, js, ts, etc.

**14/03**

**Installing Extensions in VS Code**

Creating a new Web Application

Setup File System

D:\FITNESS-APP

|

|\_public

| |\_images

| |\_docs

| |\_videos

|

|\_src

| |\_styles

| |\_scss

| |\_scripts

|

|\_package.json

|\_README.md

Reference Sites for Images:

**https://www.freepik.com/**

**https://www.canva.com/**

[**https://copilot.microsoft.com/**](https://copilot.microsoft.com/)

**Run Your Application:**

1. Click on "Go Live" in status bar of VS Code

2. Your application starts on local server with following reference address

http://127.0.0.1:5500

http://127.0.0.1:5500/public/images/banner.jpg

http://localhost:5500

**Web Page**

- Web Page is a "Hyper Text" document.

- It provides an UI (User Interface) from where user can interact with resources.

- The term "Hyper" refers to "beyond".

- A Hypertext document takes you beyond what you see on screen.

- Web Pages are classified into 2 types

a) Static Page

b) Dynamic Page

**Static Page**

- Static refers to continuous memory.

- A static page comprises of same content to send as response for every request.

- Static page have extension

.html, .htm

**Dynamic Page**

- Dynamic refers to discreet memory.

- A dynamic page customizes the response according to client request.

- Dynamic page have extension

.asp, .php, .aspx, .jsp

Note: Both static and dynamic pages are designed by using HTML.

However their composition changes

Static Page = HTML + CSS + JS + JQ

Dynamic Page = HTML + CSS + JS + JQ + Server Side Scripting

- Every web application starts with a static page "index.html".

**19/03**

**HTML**

- Hyper Text Markup Language

- Hyper means beyond.

- Hyper Text takes the user beyond the content.

- The term Markup refers to "Marking Up", which is a process of preparing our content to present on browser.

- HTML is a markup language.

- You can also call it as "Presentation Language".

**Evolution of HTML:**

- CERN labs [Council for European Research and Nuclear] developed a language for internet, which is named as "GML". [Generic Markup Language]

- GML is used for a browser called MOSAIC.

- CERN developed SGML an improved version of GML. [Standard GML]

- "Tim Berners Lee" developed HTML in early 1990's.

- HTML is superset to GML & SGML.

- IETF [Internet Engineering Task Force] developed several versions of HTML.

- IETF developed HTML up to version 3.1.

- WHATWG [Web Hyper Text Application Technology Work Group] started evolving and maintaining HTML since 2004.

- WHATWG started with HTML 4 version and now the latest till date is HTML 5. [2014]

**https://www.w3.org/ Web Standards**

**https://whatwg.org/ HTML [Evolving & Maintenance]**

**https://developer.mozilla.org/en-US/ HTML, CSS, JavaScript documents**

**Browser:**

- It is a software tool used by client to access and view web resource.

- There are various browsers

a) Edge b) Firefox c) Opera d) Safari etc..

**Browser Architecture:**

**1. User Interface**

- It refers to browser interface, which includes buttons, search bar, title bar, status, extensions, bookmarks etc.

**2. UI Backend**

- It comprises of logic required for browser UI.

- You can customize the browser UI with backed logic if it is an open source browser.

- Some popular open source browsers are

a) Chromium b) Brave c) Arora d) Dooble etc.

**3. Browser Engine**

- It translates HTML and CSS into Binary.

- Every browser have its own engine.

- Some of the popular browser engines

Edge Chakra

Firefox Spider Monkey, Gecko

Safari Webkit

Chrome V8

Opera V8

**4. Rendering Engine**

- It is responsible for presentation

- Rendering is the process of preparing the final presentation to generate as output.

**5. JavaScript Interpreter**

- It is responsible for translating JavaScript in browser.

- Interpreter translates line by line of program.

**6. Networking**

- It is responsible for locating the resources required for webpage and load the resources into browser.

- It monitors loading time and issues while loading content.

**7. Data Persistence**

- It refers to browser memory where browser can store client data.

- Browser storage includes

a) Local Storage b) Session Storage c) Cookies d) Query String etc.

**20/03**

**HTML Parsing:**

**What is HTML?**

**Evolution of HTML** - WHATWG

**Browser**

**Browser Architecture**

- UI

- UI Backend

- Browser Engine

- Rendering Engine

- JavaScript Interpreter

- Network

- Data Persistence

**HTML Parsing**

**Browser Engine:**

Developer writes Markup => Bytes => Chars => Tokens => Elements [Nodes] => DOM => Render => Layout => Paint

**FAQ's:**

1. What is Tokenization?

A. It is the process of converting the chars into tokens.

Tokens are further converted into nodes and known as elements.

**2. What is DOM?**

A. DOM refers to "Document Object Model".

It is a hierarchy of elements in page. [Parent - child nodes]

**3. What is rendering?**

A. It is the process of generating output.

**HTML Elements**

- HTML elements are classified into 5 groups

1. Normal Elements

2. Void Elements

3. RC Data Elements

4. Raw Text Elements

5. Foreign Elements

**Normal Element:**

- A normal element returns presentation directly on callback.

- It doesn't require any additional attributes.

- They start the presentation but can't stop.

- Hence a normal element requires an end token

Ex:

<b> start token

</b> end token

**Void Element:**

- The term void means no-return.

- A void element will not return any presentation directly on callback.

- It requires additional attributes to return presentation.

- After returning they stop implicitly.

- They doesn't require end token.

Ex: <img> // self ending

**RC Type Element:**

- Rich Content data elements.

- They are used only for plain text content.

- They will not allow any formats for content.

- You can't use any another element within the context of RC type.

Ex: <textarea> </textarea>

**Raw Text Element:**

- It is an HTML element that presents without any token.

- Raw text is used for presentation.

Ex:

&#8377; &copy;

**Foreign Element:**

- A foreign element can't display any presentation directly in browser.

- A browser engine can't understand and translate the element.

- It requires additional plugins to configure.

- If any element is presenting with plugin, then it is referred as "Foreign Element".

Ex: SVG, Canvas, MATHML etc.

**FAQ's**:

**1. What are block level and inline elements?**

A. Block level element will not allow any another element in same line.

Inline element allows other element side by side or in same line.

<img> inline element

<h1> block element

**2. What are generic and non-generic elements?**

A. Generic element have a predefined functionality.

Non-generic element is a static element, requires functionality to configure explicitly.

<form> generic

<p> non generic

**3. What are semantic and non-semantic elements?**

A. Semantic refers to a specific purpose and it is meant for only the given functionality.

Non Semantic acts as universal, can be used for various purpose.

<img> semantic

<div> non semantic

**21/03**

**HTML Page Structure**

**HTML Parsing**

Markup => Bytes => Chars => Tokens => Nodes [Elements] => DOM => Render => Layout => Paint

**HTML Elements**

- Normal

- Void

- RC

- Raw Text

- Foreign

Block Level

Inline

Generic

Non Generic

Semantic

Non Semantic

**HTML Page Structure**

**1. Every static page must have extension ".htm" or ".html"**

index.htm

index.html

**FAQ: What is difference between "htm & html" ?**

Ans: Technically both are same. Various software tools publish HTML pages with extension .htm as per the standards of file naming in operating system.

**2. Every web page starts with "DOCUMENT DECLARATION".**

- Document Declaration is used to define HTML version.

- It is new from HTML 5 version.

- It is defined by using the following entity

**<!DOCTYPE html>**

- If declaration is not defined then it is considered as HTML 4.

**<!doctype html>**

Note: It is always recommended to use only lowercase for token.

**3. Every web page have a document scope defined using <html>.**

Syntax:

<!DOCTYPE html>

<html>

</html>

- Scope specifies the start and end of a document in browser.

- Browser can merge multiple documents into one body.

- Hence a scope is mandatory to identify the contents of document.

**4. Every document scope must define region and language for document content.**

<!DOCTYPE html>

<html lang="en-in"> (en-in India), (en-us US) (en-gb Britan etc..) </html> - lang is a language attribute.

- "en-in" is value for language attribute.

**FAQ: What is an attribute?**

Ans: Attribute defines additional behavior for element.

**5. Every document scope comprises of 2 sections at high level.**

a) <head> b) <body>

Syntax:

<!DOCTYPE html>

<html lang="en-in">

<head>

</head>

<body>

</body>

</html>

- Content in body section loads always as a result of round-trip.

- Content in head section is cached and loaded from cache. It saves round-trip.

**FAQ: What is round-trip?**

Ans: In network application if requested content is loaded every time from server then it is known as round-trip.

**6. Head section typically comprises of elements like**

<title> , <link>, <meta>, <base>, <style>, <script>

Note: It is not mandatory to keep the elements in head section. You can change the location according to requirements.

**22/03**

**1. Document Declaration**

<!DOCTYPE html>

**2. Document Scope**

<html lang="en-in">

**3. Sections**

<head> <body>

**Elements in Head Section**

**1. Title**

- It can display page title in browser title bar.

- It can be used in bookmarking.

- It is also used in SEO. [Search Engine Optimization]

**Syntax**:

<head>

<title> Fitness | Home </title>

</head>

**FAQ: Can we set title in body section?**

Ans: Yes.

**FAQ: When to set title in body?**

Ans: If title have to change dynamically then always render using round-trip.

Body section always renders on round-trip.

**FAQ: What is the role of title in head section?**

Ans: It renders by saving round-trip.

If title is same across multiple requests then set in head section.

**2. Link**

- It can link external documents with your page.

- It can also link various plugins to page.

- External documents include CSS style sheets, favicons etc.

**Linking Favicon to page:**

1. Open MS-paint on your device

2. Go to File > Image Properties > set size 32 x 32 pixels

3. Draw your icon

4. Save into your project physical path "D:\fitness-app\public\images"

5. Name it as "favicon".

6. The default type is PNG. (favicon.png)

7. Open image location in VS Code

8. Rename the file to "favicon.ico".

9. Link to your HTML page

<head>

<link rel="shortcut icon" href="./public/images/favicon.ico">

</head>

rel : It refers to file type. [MIME type]

href : It refers to file virtual path.

**Note: Never you the physical path. Server can access only virtual path.**

/ forward slash => virtual path

\ back slash => physical path

**FAQ: What is MIME?**

Ans : MIME is "Multipurpose Internet Mail Extension".

It is the file type used by browser to identify the file content type.

**3. Meta**

- Meta refers to "Meta Data".

- It is information about your page provided to browser and SEO.

- It is responsible for making the page SEO friendly and Responsive.

**FAQ: What is Responsive?**

Ans: A response page can adjust its content according to the device.

It can also change dynamically.

**Note: You download and install "Mobile Simulator" on your browser.**

Google => Chrome Mobile Simulator

**Syntax:**

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

</head>

Ex:

<!DOCTYPE html>

<html lang="en-in">

<head>

<title>Fitness | Home</title>

<link href="./public/images/favicon.ico" rel="shortcut icon">

<meta name="viewport" content="width=device-width, initial-scale=1" >

</head>

<body>

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Unde consequuntur vel obcaecati odio impedit nemo, mollitia veniam, ratione quas expedita aperiam consectetur voluptatum architecto reiciendis qui nobis possimus vero nulla.

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Dolore inventore iusto cupiditate in molestias quaerat consectetur, odit autem dolor, iste deserunt accusantium pariatur eum excepturi reiciendis ea, quae nisi vero!

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Perferendis omnis aspernatur iure est ullam maiores, ipsum hic eos dignissimos provident vitae asperiores modi quia commodi laboriosam assumenda dolores? Quos, ea.

</body>

</html>

**24/03**

**HTML Page Structure**

**1. Document Declaration**

**2. Document Scope**

**3. Head Section Elements**

- Title

- Link

- Meta

**a) Responsive**

**b) SEO friendly**

**a) Viewport**

<meta name="viewport" content="width=device-width, initial-scale=1">

**b) Keywords**

- It refers to the words used by client to find your business.

- Keywords are usually business related terms used in SEO.

**Syntax:**

<meta name="keywords" content="gym, fitness, weight loss ..">

**c) Description**

- It is the summary of your website.

- It contains the offers and services provided by your business.

**Syntax:**

<meta name="description" content="Offers online fitness tips, provides courses for weight loss">

**d) Author**

- It is mostly used for blogs.

- It contains information about the blog author. [owner]

**Syntax:**

<meta name="author" content="author\_name">

**e) HttpEquiv**

- It defines how to handle request from client.

- It also defines when to load the content.

**Syntax:**

<meta http-equiv="refresh" content="5">

5 = 5 seconds

**f) Charset**

- It is used to identify the language used for designing page.

- Browser requires to allocate relative memory for translating the characters.

- Characters used for various languages are standardized by UTF.

[Unicode Transformation Format]

8 bit = English

16bit ]

32bit ] = for other languages with large character set

64bit ] Chinese, Korean, Arabic etc.

**Syntax:**

<meta charset="UTF-8">

Ex:

<!DOCTYPE html>

<html lang="en-in">

<head>

<title>Fitness | Home</title>

<link href="./public/images/favicon.ico" rel="shortcut icon">

<meta name="viewport" content="width=device-width, initial-scale=1" >

<meta name="keywords" content="Gym, Fitness, Diet Plan, Weight Loss">

<meta name="author" content="John">

<meta http-equiv="refresh" content="5">

<meta charset="utf-8">

</head>

</html>

**Summary:**

**1. title**

**2. link**

**3. meta**

**HTML Body Section**

- Content in body section is loaded as a result of round-trip.

- HTML <body> element provides following attributes

1. bgcolor It sets a background for page.

2. text It sets color for text in page.

**FAQ: How to define color in HTML?**

Ans: HTML colors can be configured using following techniques

**a) Color Name**

**b) Shade Name**

**c) Hexadecimal Code**

**Color Name** : It refers to exact name of color.

<body bgcolor="green">

**Shade Name** : It refers to dark and light shades of color.

<body bgcolor="lightgreen">

**Hexadecimal** : It can be 3 chars or 6 chars hexadecimal with "#".

#RGB > red, green, blue

#RRGGBB

R, G, B > 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f

<body bgcolor="red">

<body bgcolor="#f00">

<body bgcolor="#ff0000">

Ex:

<body bgcolor="red" text="#fff">

**3. Background** : It sets a background image for page.

It can't control the image size or position.

**Note: To control background image size, position and effects you need CSS background attributes.**

**CSS Background Styles:**

background-size : width & height in pixels or cover, contain

background-position : X & Y in pixels or left, right, center, top, bottom.

background-repeat : repeat, no-repeat, repeat-x, repeat-y

background-attachment : fixed, scroll

**Syntax:**

<style>

body

{

background-size: cover;

background-repeat: no-repeat;

background-attachment: scroll;

}

</style>

**Syntax:**

<body background="./public/images/banner.jpg">

</body>

**25/03**

**HTML Body Section Attributes**

**1. bgcolor**

**2. text**

**3. background**

**How to set HTML colors?**

- Hexadecimal #RGB #RRGGBB => 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f

**How to change background image style?**

background-size

background-position

background-repeat

background-attachment

**themes.getbootstrap.com**

**colorlib.com**

**4. align :** It aligns the body content left, center, right or justify.

Syntax:

<body align="left | center | right | justify">

</body>

**5. leftmargin**  ]

**6. rightmargin**  ] sets space between content and page border.

**7. topmargin**  ]

**8. bottommargin** ]

**Syntax:** <body leftmargin="100" rightmarign="100" topmargin="50" bottommargin="50"> </body>

**9. alink** : It sets color for active links in document.

**10. vlink** : It sets color for visited links in document.

**Syntax:** <body alink="red" vlink="green">

<a href="https://www.amazon.in"> Amazon </a>

</body>

**Ex:**

<!DOCTYPE html>

<html lang="en-in">

<head>

<title>Fitness | Home</title>

<link href="./public/images/favicon.ico" rel="shortcut icon">

<meta name="viewport" content="width=device-width, initial-scale=1" >

<style>

</style>

</head>

<body vlink="green" alink="red">

<a href="https://www.amazon.in">Amazon India</a> |

<a href="https://www.amazon.com">Amazon US</a> |

<a href="https://ebay.in">Ebay Shopping</a>

</body>

</html>

**Summary:**

- bgcolor

- text

- background

- align

- margins [left, right, top, bottom]

- alink

- vlink

**HTML Body Layout Design**

- HTML 4 uses table for designing body layout.

- Table leads to a situation known as "Kiss-of-Death", where it is not sure about the order of loading content.

- Content kept is table is not much SEO friendly.

- HTML 5 introduced new semantic and non-semantic elements to design body layout more SEO friendly.

- HTML 5 non-semantic elements for layout design are:

a) <div>

b) <span>

- HTML 5 semantic elements for layout design are:

a) <header>

b) <footer>

c) <section>

d) <main>

e) <nav>

f) <article>

g) <aside>

h) <dialog>

i) <address>

j) <figure>

k) <figcaption>

CSS Border Styles: border : It sets border size, style and color.

{

border: 1px solid | double | dotted red; }

**CSS Margin & Padding:**

margin : It is the space between border and page. [content outside border]

padding : It is the space between border and content inside.

margin-left

margin-right

margin-top

margin-bottom

margin [short hand for all directions]

padding-left

padding-right

padding-top

padding-bottom

padding [short hand for all directions]

**Syntax:**

<style>

div {

border: 1px solid black;

margin: 10px;

padding: 10px;

}

</style>

**CSS Font Styles:**

font-weight : It sets bold

font-style : It sets italic

font-size : It sets text size

color : It sets text color

font-family : It sets a font face [Arial, Times New Roman, ...]

**Syntax:**

span {

font-weight: bold;

font-style: italic;

color: red;

font-size: 20px;

font-family: Arial;

}

**Ex:**

Index.html

<!DOCTYPE html>

<html lang="en-in">

<head>

<title>Fitness | Home</title>

<link href="./public/images/favicon.ico" rel="shortcut icon">

<meta name="viewport" content="width=device-width, initial-scale=1" >

<style>

div {

border: 1px solid black;

margin:10px;

padding: 10px;

font-family: Arial;

}

span {

color: red;

font-size: 23px;

font-weight: bold;

font-style: italic;

font-family: Script MT;

}

</style>

</head>

<body>

<div>

Web Technologies

<div>

Front End

<div>

HTML

<p>It is a <span>markup</span> language.</p>

<p>It is used for presentation.</p>

</div>

<div>

CSS

<p>It is used for styling. </p>

</div>

</div>

<div>

Back End

<div>

Node JS

<p>It is used for creating web apps.</p>

</div>

<div>

MongoDB

<p>It is a database.</p>

</div>

</div>

</div>

</body>

</html>

**26/03**

Shopping Template

**themes.getbootstrap.com**

**https://themes.getbootstrap.com/preview/?theme\_id=37702**

Setup Icons for your project:

- Icons are provided by 3rd party libraries like

**a) Fontawesome**

**b) Bootstrap Icons**

- Implement bootstrap icons for project

**icons.getbootstrap.com**

1. Open terminal from your project

2. Run the following command

> npm install bootstrap-icons --save

3. It will add a new folder into project by name "node\_modules".

4. All library files that you install for project are kept in node\_modules.

5. Bootstrap Icons are defined using a CSS file copied into node\_modules.

6. Link the bootstrap icons CSS file to your web page.

node\_modules

|

|\_bootstrap-icons

| |

| |\_font

| |

| |\_bootstrap-icons.css

<head>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

7. Every icon in CSS file is configured with a class name. You have to access and apply icons by using class attribute.

<span class="bi bi-house"> </span>

bi bi-heart

bi bi-gift

bi bi-cart4

- get list of icons & classes from "icons.getbootstrap.com"

**Setup Icons using CDN [Content Delivery Network]:**

1. Visit "icons.getbootstrap.com"

2. Go to CDN library location [ bottom of page ]

3. Copy the CDN CSS link and paste in your page head section.

<head>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/ font /bootstrap-icons.min.css">

</head>

4. You can apply icon classes to elements

<div class="bi bi-facebook"> </div>

<span class="bi bi-twitter"> </span>

**Semantics for body layout design**

**1. <header>**

- It defines the content to display at top margin of page.

- Typically header comprises of brand name, logo, navbar, search bar, shortcuts etc.

Syntax:

<header>

... your content...

</header>

**2. <aside>**

- It is a container that comprises of content, which can move user to any sponsored or networking sites from the current website.

- It navigates aside. [outside current website]

Syntax:

<aside>

... ads.. social networking links ..

</aside>

**3. CSS Display Flex:**

display:flex It can arrange elements in a container flexible row or column wise.

flex-direction It can change direction to row or column.

justify-content It can arrange elements in container

a) baseline [left]

b) center

c) end [right]

d) space-between

e) space-around

f) space-evenly

**Syntax**:

container {

display:flex;

flex-direction: row;

justify-content: space-between;

}

**Ex: Shopper-Template**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Shopper</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

.header-row-1 {

background-color: #e6e4e4;

padding: 15px;

font-family: Arial;

font-size: 14px;

display: flex;

flex-direction: row;

justify-content: space-between;

}

.header-row-1 span {

padding-left: 10px;

padding-right: 10px;

}

.header-row-2 {

display: flex;

flex-direction: row;

justify-content: space-between;

align-items: center;

padding: 40px;

font-size: 20px;

font-family: Arial;

}

.brand-title {

font-size: 30px;

font-weight: bold;

}

nav span {

padding-left: 15px;

padding-right: 15px;

font-size: 18px;

}

.short-cuts span {

padding-left: 5px;

padding-right: 5px;

}

</style>

</head>

<body>

<header>

<div class="header-row-1">

<div>

<span class="bi bi-truck"> FREE SHIPPING WORLDWIDE </span>

<span>United States <span class="bi bi-chevron-down"></span> </span>

<span>USD <span class="bi bi-chevron-down"></span> </span>

<span>English <span class="bi bi-chevron-down"></span> </span>

</div>

<div>

<span>Shipping</span>

<span>FAQ</span>

<span>Contact</span>

</div>

<div>

<aside>

<span class="bi bi-facebook"></span>

<span class="bi bi-twitter"></span>

<span class="bi bi-instagram"></span>

<span class="bi bi-youtube"></span>

</aside>

</div>

</div>

<div class="header-row-2">

<div>

<span class="brand-title">Shopper.</span>

</div>

<div>

<nav>

<span>Home</span>

<span>Catalog</span>

<span>Shop</span>

<span>Blog</span>

<span>Pages</span>

<span>Docs</span>

</nav>

</div>

<div class="short-cuts">

<span class="bi bi-search"></span>

<span class="bi bi-person"></span>

<span class="bi bi-heart"></span>

<span class="bi bi-cart4"></span>

</div>

</div>

</header>

</body>

</html>

**27/03**

Bootstrap Icons

Body Section Semantics for Layout Design

<header>

<nav>

<aside>

**4. <section>**

- It defines the area between header and footer.

- Section comprises of all body contents.

**5. <article>**

- It is used to publish latest trends, offers, news, announcements etc.

- You can have multiple articles in a page.

- Article must contain information about current context.

Syntax:

<header> </header>

<section>

<article> </article>

</section>

<footer> </footer>

**6. <Main>**

- In computer programming main is considered as "Entry Point".

- Every program start execution from entry point.

- In a web page design we can have multiple entry point.

- It defines the area from where user can start using our website.

**Syntax:**

<main>

... your content ...

</main>

- You can configure entry point in header or section area.

**CSS Display Grid:**

- It allows to split the content into rows and columns.

- It can have a fixed set of rows and columns.

- Every page can split into maximum 12 columns. [fractions]

**Syntax:**

container {

display: grid;

grid-template-columns: 6fr 6fr;

}

**CSS Background Image:**

background-image: url("../path/name.jpg");

background-size

background-position

background-repeat

background-attachment

**CSS Text Shadow & Box Shadow**

text-shadow : horizontal\_px vertical\_px blur\_px color;

{

text-shadow: 5px 2px 1px black

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Shopper</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

.header-row-1 {

background-color: #e6e4e4;

padding: 15px;

font-family: Arial;

font-size: 14px;

display: flex;

flex-direction: row;

justify-content: space-between;

}

.header-row-1 span {

padding-left: 10px;

padding-right: 10px;

}

.header-row-2 {

display: flex;

flex-direction: row;

justify-content: space-between;

align-items: center;

padding: 40px;

font-size: 20px;

font-family: Arial;

}

.brand-title {

font-size: 30px;

font-weight: bold;

}

nav span {

padding-left: 15px;

padding-right: 15px;

font-size: 18px;

}

.short-cuts span {

padding-left: 5px;

padding-right: 5px;

}

article {

background-color: black;

color:white;

padding: 15px;

text-align: center;

font-family: Arial;

font-size: 16px;

}

.bi-lightning-fill {

color:gold;

}

main {

height: 450px;

display: grid;

grid-template-columns: 4fr 4fr 4fr;

margin-top: 10px;

}

.women-fashion {

background-image: url("./images/women-fashion.jpg");

background-size: cover;

display: flex;

flex-direction: column;

justify-content: space-around;

align-items: center;

opacity: 0.8;

}

.men-fashion {

background-image: url("./images/men-fashion.jpg");

background-size: cover;

display: flex;

flex-direction: column;

justify-content: space-around;

align-items: center;

opacity: 0.8;

}

.kids-fashion {

background-image: url("./images/kids-fashion.jpg");

background-size: cover;

display: flex;

flex-direction: column;

justify-content: space-around;

align-items: center;

opacity: 0.8;

}

.main-title {

font-family: Arial;

font-size: 50px;

font-weight: bold;

color:white;

text-shadow: 2px 2px 2px black;

}

.btn-shop {

background-color: white;

padding: 10px;

width: 140px;

font-family: Arial;

text-align: center;

box-shadow: 2px 2px 2px black;

border-radius: 5px;

}

.women-fashion:hover, .men-fashion:hover, .kids-fashion:hover {

opacity: 1;

cursor: grab;

}

.services {

display: grid;

grid-template-columns: 3fr 3fr 3fr 3fr;

padding: 20px;

margin-top: 20px;

}

.services .bi-truck, .bi-tag, .bi-lock, .bi-arrow-left-right {

color:red;

}

</style>

</head>

<body>

<header>

<div class="header-row-1">

<div>

<span class="bi bi-truck"> FREE SHIPPING WORLDWIDE </span>

<span>United States <span class="bi bi-chevron-down"></span> </span>

<span>USD <span class="bi bi-chevron-down"></span> </span>

<span>English <span class="bi bi-chevron-down"></span> </span>

</div>

<div>

<span>Shipping</span>

<span>FAQ</span>

<span>Contact</span>

</div>

<div>

<aside>

<span class="bi bi-facebook"></span>

<span class="bi bi-twitter"></span>

<span class="bi bi-instagram"></span>

<span class="bi bi-youtube"></span>

</aside>

</div>

</div>

<div class="header-row-2">

<div>

<span class="brand-title">Shopper.</span>

</div>

<div>

<nav>

<span>Home</span>

<span>Catalog</span>

<span>Shop</span>

<span>Blog</span>

<span>Pages</span>

<span>Docs</span>

</nav>

</div>

<div class="short-cuts">

<span class="bi bi-search"></span>

<span class="bi bi-person"></span>

<span class="bi bi-heart"></span>

<span class="bi bi-cart4"></span>

</div>

</div>

</header>

<section>

<article>

<span class="bi bi-lightning-fill"></span>

<span>HAPPY HOLIDAY DEALS ON EVERYTHING</span>

<span class="bi bi-lightning-fill"></span>

</article>

<main>

<div class="women-fashion">

<div class="main-title">Women</div>

<div class="btn-shop">

Shop Women <span class="bi bi-arrow-right"></span>

</div>

</div>

<div class="men-fashion">

<div class="main-title">Men</div>

<div class="btn-shop">

Shop Men <span class="bi bi-arrow-right"></span>

</div>

</div>

<div class="kids-fashion">

<div class="main-title">Kids</div>

<div class="btn-shop">

Shop Kids <span class="bi bi-arrow-right"></span>

</div>

</div>

</main>

<div class="services">

<div>

<span class="bi bi-truck"> </span> FREE SHIPPING

</div>

<div>

<span class="bi bi-arrow-left-right"> </span> FREE RETURNS

</div>

<div>

<span class="bi bi-lock"> </span> SECURE SHOPPING

</div>

<div>

<span class="bi bi-tag"> </span> OVER 10,000 STYLES

</div>

</div>

</section>

</body>

</html>

**28/03**

Ex: Bootstrap Website Header

**getbootstrap.com**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

header {

padding: 10px;

font-family: Arial;

font-size: 16px;

background-color: blueviolet;

color:white;

display: flex;

flex-direction: row;

justify-content: space-between;

align-items: center;

border-top-left-radius: 10px;

border-top-right-radius: 10px;

}

header span {

padding-left: 5px;

padding-right: 5px;

}

.bi-bootstrap-fill {

font-size: 22px;

}

.search-bar {

background-color: #620c9c;

padding: 10px;

width: 300px;

border-radius: 5px;

display: flex;

flex-direction: row;

justify-content: space-between;

}

.ctrl-button {

background-color: #1a0329;

border-radius: 2px;

padding: 2px;

}

.ctrl {

font-size: 14px;

}

nav span {

font-size: 18px;

}

</style>

</head>

<body>

<header>

<div>

<nav>

<span class="bi bi-bootstrap-fill"></span>

<span>Docs</span>

<span>Examples</span>

<span>Icons</span>

<span>Themes</span>

<span>Blog</span>

</nav>

</div>

<div>

<div class="search-bar">

<span class="bi bi-search"> Search </span>

<span class="ctrl-button"> <span class="ctrl">Ctrl</span> K </span>

</div>

</div>

<div>

<span class="bi bi-github"></span>

<span class="bi bi-twitter"></span>

<span class="bi bi-opencollective"></span>

<span> | </span>

<span> v5.3 <span class="bi bi-caret-down-fill"></span> </span>

<span> | </span>

<span class="bi bi-sun-fill"> <span class="bi bi-caret-down-fill"></span> </span>

</div>

</header>

</body>

</html>

**CSS Border Radius**

border-radius : all directions

border-top-left-radius

border-top-right-radius

border-bottom-left-radius

border-bottom-right-radius

Circle:

{

width : 100px;

height: 100px

border-radius: 100px;

}

CSS Positions:

relative : It is a position always given for parent element.

absolute : It is a position given for child element. To keep with regard to

parent content.

top ]

left ]

right ] for changing element position from specified direction.

bottom ] they are defined for absolute element.

Syntax:

parent {

position: relative;

}

child {

position: absolute;

left : 0px;

right: 0px;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

.btn {

width: 20px;

padding: 5px;

position: relative;

}

.badge {

font-size: 10px;

font-family: Arial;

width: 10px;

height: 10px;

border-radius: 10px;

background-color: red;

color:white;

text-align: center;

display: inline-block;

padding: 2px;

position: absolute;

left: 14px;

top: -2px;

}

</style>

</head>

<body>

<div class="btn bi bi-bell-fill">

<span class="badge">2</span>

</div>

</body>

</html>

**29/03**

- header

- nav

- aside

- article

- section

- main

**CSS Colors:**

**1. Color name**

{

color : green;

}

**2. Shade Name**

{

color: lightgreen;

}

**3. Hexadecimal Code [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f ]**

{

color : #0f0 (or) #00ff00;

}

**4. rgb()**

- It uses red, green and blue value range from 0 to 255.

{

color: rgb(redValue, greenValue, blueValue);

}

div {

color: rgb(0,255,0);

}

**5. rgba()**

- Red, Green, Blue and Alpha.

- Alpha sets transparent color from 0 to 1.

Syntax:

{

color: rgba(0,255,0,0.6);

}

**6. hsl()**

- It uses Hue, Saturation and Lightness.

- Hue is an angle with red, green and blue colors

0 to 120 deg => red

121 to 240 deg => green

241 to 360 deg => blue

- Saturation is gray shade from 1 to 100%

- Lightness is brightness 1 to 100%.

Syntax:

{

color: hsl(10deg, 50%, 50%);

}

**7. hsla()**

- Hue, Saturation, Lightness and Alpha.

- Alpha is between 0 to 1.

Syntax:

{

color: hsla(10deg, 50%, 50%, 0.5);

}

**8. Linear gradient & radial gradient**

- It allows to setup multiple colors.

- You can apply gradient only with "background-image" attribute.

Syntax:

{

background-image: linear-gradient(20deg, red, yellow, green);

}

Syntax:

{

background-image: linear-gradient(to top right, red 50%, green, yellow);

}

Syntax:

{

background-image: radial-gradient( red, yellow);

}

**Ex:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

height: 100vh;

background-image: radial-gradient(rgb(241, 27, 27) 50%, yellow, #00ff00);

}

</style>

</head>

<body>

</body>

</html>

**Ex: Netflix**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Netflix</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

body {

height: 100vh;

background-image: linear-gradient(rgb(0, 0, 0) 50%, rgb(222, 194, 194), rgb(0, 0, 0));

}

section {

background-image: url("./images/netflix-banner.jpg");

background-size: cover;

margin: 20px;

height: 450px;

border-radius: 40px;

border:1px solid white;

}

.shade {

background-color: rgba(0,0,0,0.6);

height: 100%;

border-radius: 40px;

text-align: center;

color:white;

}

header {

padding: 20px;

display: flex;

flex-direction: row;

justify-content: space-between;

font-family: Arial;

}

.brand-title {

font-size: 50px;

color:red;

font-weight: bold;

text-shadow: 1px 1px 2px white;

}

.language {

background-color: white;

padding: 10px;

border-radius: 10px;

}

.signin {

background-color: white;

padding: 10px;

border-radius: 10px;

margin-left: 10px;

display: inline-block;

width: 80px;

text-align: center;

}

.section-title {

font-size: 60px;

font-family: Arial;

font-weight: bold;

padding-top: 50px;

}

.section-subtitle {

font-size: 20px;

font-family: Arial;

padding-top: 20px;

}

main {

font-family: Arial;

}

.main-title{

padding-top: 20px;

padding-bottom: 40px;

}

.email {

background-color: rgba(126, 125, 125, 0.744);

padding: 20px;

display: inline-block;

width: 300px;

text-align: left;

border:1px solid white;

border-radius: 30px;

font-size: 20px;

}

.btn-start {

background-color: red;

padding: 20px;

display: inline-block;

width: 170px;

border-radius: 30px;

font-size: 20px;

margin-left: 10px;

}

</style>

</head>

<body>

<header>

<div>

<span class="brand-title">NETFLIX</span>

</div>

<div>

<span class="language"><span class="bi bi-translate"> Language <span class="bi bi-caret-down-fill"></span> </span></span>

<span class="signin">Sign In</span>

</div>

</header>

<section>

<div class="shade">

<div class="section-title">

Unlimited movies, TV<br> shows and more

</div>

<div class="section-subtitle">

Starts at ₹149. Cancel at any time.

</div>

<main>

<div class="main-title">

Ready to watch? Enter your email to create or restart your membership.

</div>

<div>

<span class="email"> Email Address</span>

<span class="btn-start"> Get Started <span class="bi bi-chevron-right"></span> </span>

</div>

</main>

</div>

</section>

</body>

</html>

**1/04**

====

Summary

<header>

<section>

<nav>

<aside>

<main>

<article>

**Footer**

- Footer is a container used to display content at bottom margin of page.

- Typically footer comprises of services, contact details, copyrights etc.

Syntax:

<footer>

... your content ...

</footer>

**Address**

- It is a container for contact details.

- It keeps your contact details SEO friendly.

Syntax:

<address>

... phone, email, skype etc..

</address>

**CSS Child & Adjacent Selectors:**

nav span { } all span inside nav.

<nav>

<span></span>

<span> </span>

</nav>

nav + span { } It refers to span adjacent to nav.

It refers only first element.

<nav> </nav>

<span> </span> => only first is accessed with " + "

<span> </span>

nav ~ span { } It refers to all span elements after nav.

<nav> </nav>

<span> </span> => all spans but in same scope

<span> </span>

**Ex: Shopper Template - Footer**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Shopper</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

.header-row-1 {

background-color: #e6e4e4;

padding: 15px;

font-family: Arial;

font-size: 14px;

display: flex;

flex-direction: row;

justify-content: space-between;

}

.header-row-1 span {

padding-left: 10px;

padding-right: 10px;

}

.header-row-2 {

display: flex;

flex-direction: row;

justify-content: space-between;

align-items: center;

padding: 40px;

font-size: 20px;

font-family: Arial;

}

.brand-title {

font-size: 30px;

font-weight: bold;

}

nav span {

padding-left: 15px;

padding-right: 15px;

font-size: 18px;

}

.short-cuts span {

padding-left: 5px;

padding-right: 5px;

}

article {

background-color: black;

color:white;

padding: 15px;

text-align: center;

font-family: Arial;

font-size: 16px;

}

.bi-lightning-fill {

color:gold;

}

main {

height: 450px;

display: grid;

grid-template-columns: 4fr 4fr 4fr;

margin-top: 10px;

}

.women-fashion {

background-image: url("./images/women-fashion.jpg");

background-size: cover;

display: flex;

flex-direction: column;

justify-content: space-around;

align-items: center;

opacity: 0.8;

}

.men-fashion {

background-image: url("./images/men-fashion.jpg");

background-size: cover;

display: flex;

flex-direction: column;

justify-content: space-around;

align-items: center;

opacity: 0.8;

}

.kids-fashion {

background-image: url("./images/kids-fashion.jpg");

background-size: cover;

display: flex;

flex-direction: column;

justify-content: space-around;

align-items: center;

opacity: 0.8;

}

.main-title {

font-family: Arial;

font-size: 50px;

font-weight: bold;

color:white;

text-shadow: 2px 2px 2px black;

}

.btn-shop {

background-color: white;

padding: 10px;

width: 140px;

font-family: Arial;

text-align: center;

box-shadow: 2px 2px 2px black;

border-radius: 5px;

}

.women-fashion:hover, .men-fashion:hover, .kids-fashion:hover {

opacity: 1;

cursor: grab;

}

.services {

display: grid;

grid-template-columns: 3fr 3fr 3fr 3fr;

padding: 20px;

margin-top: 20px;

}

.services .bi-truck, .bi-tag, .bi-lock, .bi-arrow-left-right {

color:red;

}

footer {

background-color: black;

color:white;

font-family: Arial;

padding: 10px;

}

.footer-row-1 {

padding-top: 40px;

padding-bottom: 40px;

text-align: center;

}

.footer-title {

font-size: 30px;

font-weight: bold;

padding-bottom: 30px;

}

.email {

background-color: rgba(135, 131, 131, 0.653);

border:1px solid gray;

padding: 20px;

display: inline-block;

width: 300px;

text-align: left;

}

.subscribe {

background-color: rgb(123, 122, 122);

padding: 20px;

width: 150px;

display: inline-block;

color:white;

margin-left: 10px;

}

.footer-row-2 {

margin-top: 30px;

display: grid;

grid-template-columns: 2.4fr 2.4fr 2.4fr 2.4fr 2.4fr;

padding: 20px;

}

.service-title {

font-size: 16px;

font-weight: bold;

text-transform: uppercase;

display: block;

padding-bottom: 20px;

}

.service-title~span {

display: block;

padding-bottom: 10px;

}

address span {

display: block;

padding-bottom: 10px;

}

aside span {

padding-right: 5px;

}

.footer-brand-title {

font-size: 30px;

font-weight: bold;

padding-bottom: 20px;

display: block;

}

</style>

</head>

<body>

<header>

<div class="header-row-1">

<div>

<span class="bi bi-truck"> FREE SHIPPING WORLDWIDE </span>

<span>United States <span class="bi bi-chevron-down"></span> </span>

<span>USD <span class="bi bi-chevron-down"></span> </span>

<span>English <span class="bi bi-chevron-down"></span> </span>

</div>

<div>

<span>Shipping</span>

<span>FAQ</span>

<span>Contact</span>

</div>

<div>

<aside>

<span class="bi bi-facebook"></span>

<span class="bi bi-twitter"></span>

<span class="bi bi-instagram"></span>

<span class="bi bi-youtube"></span>

</aside>

</div>

</div>

<div class="header-row-2">

<div>

<span class="brand-title">Shopper.</span>

</div>

<div>

<nav>

<span>Home</span>

<span>Catalog</span>

<span>Shop</span>

<span>Blog</span>

<span>Pages</span>

<span>Docs</span>

</nav>

</div>

<div class="short-cuts">

<span class="bi bi-search"></span>

<span class="bi bi-person"></span>

<span class="bi bi-heart"></span>

<span class="bi bi-cart4"></span>

</div>

</div>

</header>

<section>

<article>

<span class="bi bi-lightning-fill"></span>

<span>HAPPY HOLIDAY DEALS ON EVERYTHING</span>

<span class="bi bi-lightning-fill"></span>

</article>

<main>

<div class="women-fashion">

<div class="main-title">Women</div>

<div class="btn-shop">

Shop Women <span class="bi bi-arrow-right"></span>

</div>

</div>

<div class="men-fashion">

<div class="main-title">Men</div>

<div class="btn-shop">

Shop Men <span class="bi bi-arrow-right"></span>

</div>

</div>

<div class="kids-fashion">

<div class="main-title">Kids</div>

<div class="btn-shop">

Shop Kids <span class="bi bi-arrow-right"></span>

</div>

</div>

</main>

<div class="services">

<div>

<span class="bi bi-truck"> </span> FREE SHIPPING

</div>

<div>

<span class="bi bi-arrow-left-right"> </span> FREE RETURNS

</div>

<div>

<span class="bi bi-lock"> </span> SECURE SHOPPING

</div>

<div>

<span class="bi bi-tag"> </span> OVER 10,000 STYLES

</div>

</div>

</section>

<footer>

<div class="footer-row-1">

<div class="footer-title">Want style Ideas and Treats?</div>

<div>

<span class="email">Enter Email\*</span>

<span class="subscribe">Subscribe</span>

</div>

</div>

<div class="footer-row-2">

<div>

<span class="footer-brand-title">Shopper.</span>

<aside>

<span class="bi bi-facebook"></span>

<span class="bi bi-twitter"></span>

<span class="bi bi-instagram"></span>

<span class="bi bi-youtube"></span>

<span class="bi bi-linkedin"></span>

</aside>

</div>

<div>

<span class="service-title">Support</span>

<span>Contact Us</span>

<span>FAQs</span>

<span>Size Guide</span>

<span>Shipping & Returns</span>

</div>

<div>

<span class="service-title">Shop</span>

<span>Men's Shopping</span>

<span>Women's Shopping</span>

<span>Kids' Shopping</span>

<span>Discounts</span>

</div>

<div>

<span class="service-title">Company</span>

<span>Our Story</span>

<span>Careers</span>

<span>Terms & Conditions</span>

<span>Privacy & Cookie policy</span>

</div>

<div>

<span class="service-title">CONTACT</span>

<address>

<span>1-202-555-0105</span>

<span>1-202-555-0106</span>

<span>help@shopper.com</span>

</address>

</div>

</div>

</footer>

</body>

</html>

Ex: Flipkart footer - layout

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

footer {

background-color: black;

color:white;

font-size: 20px;

padding: 10px;

display: grid;

grid-template-columns: 6fr 6fr;

height: 300px;

}

.left-panel {

display: grid;

grid-template-columns: 3fr 3fr 3fr 3fr;

}

.right-panel {

display: grid;

grid-template-columns: 6fr 6fr;

border-left: 1px solid white;

padding-left: 20px;

}

</style>

</head>

<body>

<footer>

<div class="left-panel">

<div>Col-1</div>

<div>Col-2</div>

<div>Col-3</div>

<div>Col-4</div>

</div>

<div class="right-panel">

<div>Col-1</div>

<div>Col-2</div>

</div>

</footer>

</body>

</html>

**Dialog**

- It opens a popup window inside page.

- It is used as window with content in backdrop.

- Dialog is default hidden, you have to display with "open" attribute.

Syntax:

<dialog open>

... your content ...

</dialog>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

.dialog-header {

background-color: black;

color:white;

padding: 10px;

display: flex;

justify-content: space-between;

}

dialog {

width: 300px;

height: 400px;

padding: 30px;

box-shadow: 4px 3px 2px gray;

}

.dialog-body {

height: 250px;

margin-top:20px;

}

.text-area {

height: 250px;

border:1px solid gray;

}

.dialog-footer {

margin-top: 50px;

}

.btn-post {

width: 100%;

background-color: goldenrod;

color:white;

padding: 5px;

text-align: center;

}

</style>

</head>

<body>

<dialog open>

<div class="dialog-header">

<span>Ask Disha 2.0</span>

<span class="bi bi-x-lg"></span>

</div>

<div class="dialog-body">

<span>Your Comments</span>

<div class="text-area">

</div>

</div>

<div class="dialog-footer">

<div class="btn-post">Post Comment</div>

</div>

</dialog>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi exercitationem ex esse qui dolorem. Inventore, veniam dolores, dolorum at aliquid expedita deleniti dicta maiores sunt fugit vero, id nesciunt dolorem.

Lorem, ipsum dolor sit amet consectetur adipisicing elit. At dicta sunt cumque accusamus, magnam, mollitia amet neque, dolorem modi eum veritatis. Ipsum, delectus voluptate odit facilis rerum dignissimos ullam ipsam?

</body>

</html>

Ex: Dialog

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

dialog {

border:none;

padding: 0px;

border-top-left-radius: 20px;

border-top-right-radius: 20px;

border-bottom-left-radius: 20px;

border-bottom-right-radius: 20px;

width: 500px;

}

.dialog-header {

background-color: orangered;

color:white;

height: 100px;

padding: 20px;

font-size: 50px;

font-family: Arial;

border-radius: 20px;

display: flex;

justify-content: space-between;

}

.dialog-body {

background-color: white;

border-radius: 20px;

margin-top: -40px;

padding: 50px;

font-size: 16px;

font-family: Arial;

height: 250px;

}

.dialog-footer {

padding: 20px;

border-top: 1px solid gray;

font-size: 14px;

color:rgb(180, 180, 180);

}

body {

background-color: black;

}

</style>

</head>

<body>

<dialog open>

<div class="dialog-header">

<span class="bi bi-person-circle"> <span>Naresh IT</span> </span>

<span class="bi bi-x-circle-fill"></span>

</div>

<div class="dialog-body">

<div> Naresh IT</div>

<div>Hello , Good Afternoon</div>

<div>

Appear that you are visiting our pages! Can we help you find any suitable course ?

<br>

How may I help you ?

</div>

</div>

<div class="dialog-footer">

<span>Type your message and hit enter</span>

</div>

</dialog>

</body>

</html>

2/04

====

**Body Semantics and Entities:**

**1. Line Break <br>**

FAQ: What is difference between <br> & <br /> ?

Ans: There is no <br /> in HTML.

<br/> is just a reference given for self ending elements.

It is required for editor that can't understand void elements.

**2. Blank Spaces &nbsp; [non-breakable space]**

Syntax:

H &nbsp; &nbsp; T M L

**3. Preformatted <pre>** It keeps the formats as defined in source.

**4. Code <code>** It makes the block browser friendly.

It defines that content is a code block.

**5. Variable <var>** It is used to mark variables in programming.

**6. Sample Text <samp>** It is used to mark sample output.

**7. Large Text <big>**

**8. Smaller Text <small>**

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

var {

color:blue;

}

samp {

color:gray;

}

.output {

font-weight: bold;

}

</style>

</head>

<body>

<div>

normal <small>small</small> <big>Large</big>

</div>

<div>

<code>

<pre>

var <var>createCounter</var> = function(n){

return function(){

};

};

</pre>

</code>

</div>

<div>

<span class="output">Output :</span> <samp> [-2, -1, 0, 1, 2] </samp>

</div>

</body>

</html>

**9. Headings <h1> to <h6>**

FAQ's:

**1. Why you need an heading element, if same styles can be defined without heading?**

A. To make the topics in page SEO friendly.

**2. Can we modify the default style defined for heading?**

A. Yes.

Syntax:

h1 {

font-size : 40px;

color:red;

}

**3. What are the default styles defined for heading element?**

A. Heading element is applied with

a) font-size

b) font-weight

c) display: block

**4. How to remove any default style defined for element?**

A. By using CSS inheritance value "unset".

Syntax:

h1 { h1 {

font-size: unset; all : unset;

font-weight:unset; (or) }

display:unset;

}

**Note:**

1. Don't use heading to highlight any word or sentence in a paragraph.

2. Don't use too many headings in a page. It may SPAM your page.

**3/04**

====

- line break <br>

- blank space &nbsp;

- pre formatted <pre>

- code <code>

- variables <var>

- sample text <samp>

- big and small text <big> <small>

- headings <h1> ... <h6>

**10. Paragraphs & Blockquotes**

<p> for paragraph

<blockquote> summarizing content

- Paragraph element keep the content relative to others topics in a page.

- It provides a container to align and set padding or margin.

- Blockquote is similar to paragraph but it keeps the summary SEO friendly.

Syntax:

<p align="center">

<blockquote align="justify">

FAQ's:

**1. How to set first-line indent for paragraph?**

A. By using CSS "text-indent" attribute.

Syntax:

p, blockquote {

text-indent : 100px;

}

**2. How to set spacing between line, words and chars?**

A. By using CSS attributes

a) letter-spacing

b) word-spacing

c) line-height [ line space ]

Syntax:

p, blockquote {

letter-spacing : 2px;

word-spacing : 10px;

line-height : 50px;

}

**3. How to justify alignment without using text-align?**

A. By using CSS "word-break" set to "break-all".

Syntax:

p, blockquote {

word-break : break-all;

}

**4. How to set a drop cap?**

A. You can access paragraph first letter by using CSS "::first-letter" class.

You can make it dropped by few lines by using "float: left" attribute.

Syntax:

p ::first-letter {

font-size: 40px;

float: left;

line-height: 40px;

font-family: Algerian;

}

**5. How to create scrollable paragraph content?**

A. By using CSS "overflow" attribute set to "auto or scroll".

You have to control overflow for content in a container.

Syntax:

p {

width : 300px;

height: 200px;

padding: 10px;

border: 2px solid gray;

overflow: auto;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

blockquote {

padding-left: 300px;

padding-right: 300px;

font-style: italic;

line-height: 20px;

}

blockquote::first-letter {

font-size: 60px;

float: left;

line-height: 40px;

font-family: Magneto;

padding-right: 10px;

}

.first-para {

text-indent: 100px;

}

h2 {

letter-spacing: 10px;

word-spacing: 40px;

}

p {

word-break: break-all;

}

.terms {

width: 500px;

padding: 10px;

border:2px solid gray;

height: 200px;

overflow: auto;

}

</style>

</head>

<body>

<h2 align="center">Men Fashion</h2>

<blockquote align="justify">

Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio repellendus voluptates, unde assumenda modi mollitia tenetur quod laboriosam veritatis quis magni similique expedita, officiis perspiciatis, nobis molestias quia iste ullam? Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo reprehenderit cupiditate molestias ipsum, suscipit doloribus eaque. Nisi suscipit nemo aperiam dolorum, incidunt error fuga repellat quam placeat, dolorem laborum sit.

</blockquote>

<p class="first-para">

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Itaque minima perspiciatis similique sapiente! A necessitatibus error quam esse quo doloribus magnam velit deserunt natus aliquam! Ab exercitationem magnam tempora dicta? Lorem ipsum dolor sit amet consectetur adipisicing elit. Dicta consectetur quos in quam minus atque possimus, omnis debitis vero, eligendi amet dolorum eaque, magnam magni modi veniam. Soluta, nemo voluptate?

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Est nulla dignissimos quidem, accusantium minus laboriosam voluptatem odio. Earum ullam consequatur reprehenderit quas nesciunt quos, facere obcaecati, voluptate aliquam animi ad.

</p>

<h3>Terms of Service</h3>

<div class="terms">

<p>

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Itaque minima perspiciatis similique sapiente! A necessitatibus error quam esse quo doloribus magnam velit deserunt natus aliquam! Ab exercitationem magnam tempora dicta? Lorem ipsum dolor sit amet consectetur adipisicing elit. Dicta consectetur quos in quam minus atque possimus, omnis debitis vero, eligendi amet dolorum eaque, magnam magni modi veniam. Soluta, nemo voluptate?

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Est nulla dignissimos quidem, accusantium minus laboriosam voluptatem odio. Earum ullam consequatur reprehenderit quas nesciunt quos, facere obcaecati, voluptate aliquam animi ad.

</p>

<p>

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Itaque minima perspiciatis similique sapiente! A necessitatibus error quam esse quo doloribus magnam velit deserunt natus aliquam! Ab exercitationem magnam tempora dicta? Lorem ipsum dolor sit amet consectetur adipisicing elit. Dicta consectetur quos in quam minus atque possimus, omnis debitis vero, eligendi amet dolorum eaque, magnam magni modi veniam. Soluta, nemo voluptate?

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Est nulla dignissimos quidem, accusantium minus laboriosam voluptatem odio. Earum ullam consequatur reprehenderit quas nesciunt quos, facere obcaecati, voluptate aliquam animi ad.

</p>

</div>

</body>

</html>

**6. How to set ellipsis for paragraph?**

A. You have to display paragraph content in one line by removing word-wrap.

{

white-space: nowrap;

}

Hide the overflowing paragraph.

Set text-overflow to ellipsis.

Syntax:

p {

width : 350px;

border: 1px solid gray;

white-space: nowrap;

overflow: hidden;

text-overflow: ellipsis;

}

Note: You can set screen-tip for any content in page by using "title" attribute.

It is a HTML attribute defined for element to display mouse over tips.

<h1 title="Hyper Text Markup Language"> HTML </h1>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

p {

width: 350px;

border:2px solid gray;

padding: 5px;

white-space: nowrap;

overflow: hidden;

text-overflow: ellipsis;

}

</style>

</head>

<body>

<h2 title="Hyper Text Markup Language">HTML</h2>

<p title="Lorem ipsum dolor sit, amet consectetur adipisicing elit. Quis, saepe. Maiores voluptate eveniet natus velit, alias reprehenderit maxime asperiores magnam nulla ipsa perspiciatis dignissimos error qui nostrum quas? Fugiat, laboriosam?">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Quis, saepe. Maiores voluptate eveniet natus velit, alias reprehenderit maxime asperiores magnam nulla ipsa perspiciatis dignissimos error qui nostrum quas? Fugiat, laboriosam?</p>

</body>

</html>

**7. How to display paragraphs in multiple columns?**

A. By using CSS "columns" attribute or CSS display "grid".

- Display Grid is a disconnected grid system.

- Columns is a connected grid system. [ the content of one column spans to another column when it reaches the bottom margin of column].

a) columns

b) column-width

c) column-gap

d) column-rule

Syntax:

.container {

columns : 5;

column-gap : 10px;

column-rule: 1px dotted gray;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.block {

border:1px solid black;

padding: 5px;

font-family: Arial Black;

font-size: 16px;

width: 180px;

}

.split {

color:aquamarine;

}

.shift {

color:gray;

}

.title {

font-family: Arial Black;

font-weight: bold;

font-size: 18px;

padding-left: 10px;

}

.title-block {

display: flex;

}

.sub-title {

font-size: 43px;

font-family: Arial;

}

.news {

columns: 4;

column-gap: 15px;

column-rule: 1px dotted gray;

font-size: 12px;

text-align: justify;

margin-top: 20px;

}

.news-block {

box-shadow: 4px 3px 2px black;

padding: 10px;

border:1px solid black;

width: 800px;

}

</style>

</head>

<body>

<div class="news-block">

<div class="title-block">

<div class="block">

<span>SHARP</span>

<span class="split">|</span>

<span class="shift">SHIFT</span>

</div>

<div class="title">Despite US companies' preferences, Inidian community faces ongoing uncertanity</div>

</div>

<div class="sub-title">

Indian upset over reduces H-1B visa pool

</div>

<div class="news">

<h4>Neeraj Kumar | DC | Delhi</h4>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsam libero molestias distinctio cum dolores recusandae provident doloribus reprehenderit quasi ut Lorem ipsum dolor, sit amet consectetur adipisicing elit. Architecto corporis dignissimos molestiae harum dolorem sed temporibus officiis velit culpa. Pariatur quo nesciunt et ipsum sint cumque doloribus ex. Atque, nisi! et, deleniti mollitia laborum nostrum dolorum possimus fugiat in aliquam! Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusamus reiciendis amet fugit dicta quasi sit facilis exercitationem doloribus quisquam modi non maxime consectetur, iusto itaque laudantium adipisci architecto inventore nisi? Lorem ipsum dolor sit amet consectetur adipisicing elit. Qui assumenda tempore dolor dolorem consectetur saepe harum est, dolorum, molestiae officia veritatis repellendus deserunt mollitia fugit molestias, at vero minus impedit?</p>

<p>Lorem ipsum dolor Lorem ipsum dolor sit, amet consectetur adipisicing elit. Nam nisi odit nesciunt? Dolore et nulla praesentium quam eligendi expedita eaque corrupti error quia ipsum nam suscipit, incidunt deleniti sequi! Odio. sit amet consectetur adipisicing elit. Pariatur impedit, laudantium in, nam nisi molestias illum magnam repellendus placeat reiciendis velit earum nihil quibusdam soluta sed iste. Doloribus, odio minus? Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae nisi ab aliquid recusandae enim fugiat est quae cum modi dolores. Sit quae, eveniet accusantium eius maiores ipsum nobis minima ab.</p>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Pariatur impedit, laudantium in, nam nisi molestias illum magnam repellendus placeat reiciendis velit earum nihil quibusdam soluta sed iste. Doloribus, odio minus? Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae nisi ab aliquid recusandae enim fugiat est quae cum modi dolores. Sit quae, eveniet accusantium eius maiores ipsum nobis minima ab.</p>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Pariatur impedit, laudantium in, nam nisi molestias illum magnam repellendus placeat reiciendis velit earum nihil quibusdam soluta sed iste. Doloribus, odio minus? Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae nisi ab aliquid recusandae enim fugiat est quae cum modi dolores. Sit quae, eveniet accusantium eius maiores ipsum nobis minima ab.</p>

</div>

</div>

</body>

</html>

**4/04**

====

**Lists in HTML**

**1. Data List with terms and definitions**

<dl> data list

<dt> data term

<dd> data definition

Syntax:

<dl>

<dt> Term </dt>

<dd> Definition </dd>

</dl>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

dt {

font-weight: bold;

font-family: Arial;

background-color: black;

color:white;

padding: 4px;

margin-bottom: 2px;

}

dl {

width: 300px;

}

dd {

padding: 4px;

margin-bottom: 5px;

background-color: lightgray;

}

</style>

</head>

<body>

<h2>Web Technologies</h2>

<dl>

<dt>HTML</dt>

<dd>It is a markup language.</dd>

<dd>It is used for presentation.</dd>

<dt>JavaScript</dt>

<dd>It manipulates DOM. </dd>

<dt>jQuery </dt>

<dt>Library</dt>

<dd>It a JavaScript library for DOM. </dd>

</dl>

</body>

</html>

- You can split terms and definitions into columns using "display: grid".

- But for grid style every term can have only one definition.

Syntax:

dl {

display: grid;

grid-template-columns: 3fr 9fr;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

dt {

font-weight: bold;

font-family: Arial;

background-color: black;

color:white;

padding: 2px;

margin-bottom: 2px;

}

dl {

display: grid;

grid-template-columns: 3fr 9fr;

}

dd {

padding: 2px;

margin-bottom: 5px;

}

.control {

width: 200px;

height: 25px;

background-color: white;

border:1px solid black;

padding: 2px;

}

</style>

</head>

<body>

<dl>

<dt>User Name</dt>

<dd><div class="control">John</div></dd>

<dt>Password</dt>

<dd><div class="control"> \*\*\*\* </div></dd>

<dt>Date of Birth</dt>

<dd><div class="control"></div></dd>

<dt>City</dt>

<dd><div class="control"></div></dd>

</dl>

</body>

</html>

**CSS Sticky Position:**

- It allows the element to scroll along with content.

- It can lock scrolling and stick to page at specified location.

Syntax:

element {

position: sticky;

top:0px;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

nav {

width: 150px;

}

dt {

font-weight: bold;

background-color: black;

color:white;

padding: 2px;

margin-bottom: 10px;

position: sticky;

top: 0px;

}

dd {

background-color: gray;

color:white;

padding: 2px;

margin-bottom: 10px;

}

</style>

</head>

<body>

<nav>

<dl>

<h2>Amazon</h2>

<dt>Electronics</dt>

<dd>Televisions</dd>

<dd>Mobiles</dd>

<dd>Watches</dd>

<dd>Televisions</dd>

<dd>Mobiles</dd>

<dd>Watches</dd>

<dd>Televisions</dd>

<dd>Mobiles</dd>

<dd>Watches</dd>

<dd>Televisions</dd>

<dd>Mobiles</dd>

<dd>Watches</dd>

<dd>Televisions</dd>

<dd>Mobiles</dd>

<dd>Watches</dd>

<dd>Televisions</dd>

<dd>Mobiles</dd>

<dd>Watches</dd>

<dd>Televisions</dd>

<dd>Mobiles</dd>

<dd>Watches</dd>

<dt>Fashion</dt>

<dd>Kids</dd>

<dd>Women</dd>

<dd>Men</dd>

<dd>Kids</dd>

<dd>Women</dd>

<dd>Men</dd>

<dd>Kids</dd>

<dd>Women</dd>

<dd>Men</dd>

<dd>Kids</dd>

<dd>Women</dd>

<dd>Men</dd>

<dd>Kids</dd>

<dd>Women</dd>

<dd>Men</dd>

<dd>Kids</dd>

<dd>Women</dd>

<dd>Men</dd>

<dd>Kids</dd>

<dd>Women</dd>

<dd>Men</dd>

<dd>Kids</dd>

<dd>Women</dd>

<dd>Men</dd>

<dd>Kids</dd>

<dd>Women</dd>

<dd>Men</dd>

<dt>Footwear</dt>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

<dd>Casuals</dd>

<dd>Sneakers</dd>

<dd>Boots</dd>

</dl>

</nav>

</body>

</html>

**2. Ordered List**

- It sets auto numbering for a list of items.

- Numbering updates automatically when you add or remove item.

<ol> Ordered list

<li> List item

Syntax:

<ol>

<li>Item-1</li> 1. Item-1

<li>Item-2</li> 2. Item-2

</ol>

- You can change the numbering style by using "type" attribute set with

"1, A, a, i, I".

Syntax:

<ol type="A">

<ol type="i">

<ol> => default type is 1.

- You can set startup numbering level using "start" attribute.

- It refers to level number to start with. It is always a number value.

Syntax:

<ol type="1" start="5">

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

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

- You can use "reversed" attribute to set reverse numbering order.

- It will not reverse the items, it just reverse order of numbering.

Syntax:

<ol type="1" reversed>

<ol type="a" reversed>

- You can create a nested numbering list.

- But make sure that the child list is inside <li> Element.

[not below the <li> element]

Syntax: Bad Code

<li> Parent </li>

<ol>

<li> Child </li>

</ol>

Syntax: Good Code

<li> Parent

<ol>

<li>Child</li>

</ol>

</li>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<ol type="1">

<li>HTML

<ol type="a">

<li>Semantic Elements

<ol type="i">

<li>Heading</li>

<li>Header</li>

</ol>

</li>

<li>Generic Elements</li>

</ol>

</li>

<li>JavaScript

<ol type="a">

<li>Variables</li>

<li>Data Types</li>

</ol>

</li>

</ol>

</body>

</html>

FAQ's:

**1. How to display the list items side by side?**

A. By using display attribute set to "flex" for list not list item.

Syntax:

ol {

display : flex;

}

As default direction is row. It keeps elements side by side.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.child-list {

display: flex;

flex-direction: row;

margin-top: 10px;

}

li {

margin-left: 20px;

margin-right: 20px;

}

</style>

</head>

<body>

<ol type="1">

<li>\_\_\_\_\_\_\_\_\_\_ elements doesn't require end token.

<ol type="a" class="child-list">

<li>Normal</li>

<li>RC Data</li>

<li>Void</li>

<li>Raw Text</li>

</ol>

</li>

</ol>

</body>

</html>

**2. How to display list items in multiple columns?**

A. By using Grid style or by using columns.

Syntax:

ol {

display: grid;

grid-template-columns: 6fr 6fr;

}

ol {

columns : 2;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.child-list {

columns: 2;

margin-top: 10px;

}

</style>

</head>

<body>

<ol type="1">

<li>\_\_\_\_\_\_\_\_\_\_ elements doesn't require end token.

<ol type="a" class="child-list">

<li>Normal</li>

<li>RC Data</li>

<li>Void</li>

<li>Raw Text</li>

</ol>

</li>

</ol>

</body>

</html>

**3. How to remove numbering for list items?**

A. By using CSS "list-style" attribute set to "none".

Syntax:

ol {

list-style : none;

}

CSS Before & After Selectors:

::before It keeps a content before specified item

::after It keeps content after specified item

Syntax:

element :: before {

content : " any ";

}

Ex: Breadcrumb Design

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

ol {

list-style: none;

display: flex;

}

li::after {

content: " > ";

padding-left: 5px;

color:gray;

}

li::before {

content: " ";

padding-right: 5px;

}

</style>

</head>

<body>

<nav>

<ol>

<li>Home</li>

<li>Mobiles & Accessories</li>

<li>Mobiles</li>

<li>Apple Mobile</li>

<li>Apple 16</li>

</ol>

</nav>

</body>

</html>

**05/04**

=====

**Unordered List:**

- It is a bulleted list.

- List items are defined with a bullet symbol.

- It supports bullet with type as "circle, disc & square".

<ul> Unordered List

<li> List Item

Syntax:

<ul>

<li> Item-1 </li>

<li> Item-2 </li>

</ul>

<ul type="circle | disc | square">

</ul>

**FAQ: How to set custom bullet symbol?**

Ans: You can use bootstrap icons or you can set custom image by using the CSS

attribute "list-style-image".

Syntax:

<li class="bi bi-house"> Home </li>

Syntax:

ul {

list-style: none;

list-style-image: url("./image/pic.gif");

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

ul {

list-style: none;

list-style-image: url("./images/bullet.gif");

font-size: 30px;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body>

<ul>

<li> HTML</li>

<li> CSS</li>

<li> JavaScript </li>

</ul>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

ul {

list-style: none;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body>

<ul>

<li class="bi bi-person-fill"> Sign In </li>

<li class="bi bi-gift"> Offers </li>

<li class="bi bi-translate"> Language </li>

<li class="bi bi-envelope-fill"> Email </li>

</ul>

<ul>

<li class="bi bi-tag-fill">Axis Bank Offer 20% OFF</li>

<li class="bi bi-tag-fill">HDFC Bank Offer 100 Cash Back</li>

<li class="bi bi-tag-fill">IDFC zero cost EMI Offer</li>

</ul>

</body>

</html>

**FAQ: Can we set bullet symbol for ordered list and vice versa?**

Ans: Yes. By using CSS attribute "list-style-type" you can change symbol to number or

vice versa.

Syntax:

ol {

list-style-type : circle;

}

ul {

list-style-type : decimal | lower-alpha | any ;

}

Note: You can configure a nested list with combination of Ordered and Unordered

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body>

<ul>

<li> HTML

<ol type="a">

<li>Normal Elements</li>

<li>Void Elements</li>

</ol>

</li>

<li> CSS</li>

<li> JavaScript </li>

</ul>

</body>

</html>

**Details & Summary**

- Details is a container with expandable and collapsible content.

- Summary is a caption of details container.

Syntax:

<details>

<summary> Caption </summary>

.... your content....

</details>

- If you want the details to keep open the content then apply "open" attribute.

Syntax:

<details open>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

details div {

margin-left: 20px;

margin-bottom: 10px;

margin-top: 5px;

}

</style>

</head>

<body>

<details>

<summary> Need Help in Sign in? </summary>

<div>Forgot Password</div>

<div>Recover Id</div>

<div>Other Issues</div>

</details>

<details open>

<summary>Electronics</summary>

<div>Televisions</div>

<div>Mobiles</div>

<div>Watches</div>

</details>

</body>

</html>

Fieldset & Legend

- Fieldset is a container with frame.

- It is used to keep relative content together.

- Legend is a caption set of fieldset.

Syntax:

<fieldset>

<legend> Caption </legend>

.... your content....

</fieldset>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.control {

border:1px solid black;

width: 200px;

height: 22px;

margin-bottom: 5px;

background-color: white;

}

dl {

display: grid;

grid-template-columns: 2fr 8fr;

}

fieldset {

margin: 20px;

width: 500px;

box-shadow: 4px 4px 4px black;

background-color: rgb(200, 238, 238);

border-radius: 10px;

}

legend {

background-color:rgb(1, 113, 113);

color:white;

padding: 4px;

width: 130px;

text-align: center;

border-radius: 10px;

box-shadow: 2px 2px 2px gray;

}

</style>

</head>

<body>

<fieldset>

<legend>Personal Details</legend>

<dl>

<dt>First Name</dt>

<dd class="control"></dd>

<dt>Last Name</dt>

<dd class="control"></dd>

<dt>Date of Birth</dt>

<dd class="control"></dd>

</dl>

</fieldset>

<fieldset>

<legend>Contact Details</legend>

<dl>

<dt>City</dt>

<dd class="control"></dd>

<dt>State</dt>

<dd class="control"></dd>

<dt>Postal Code</dt>

<dd class="control"></dd>

</dl>

</fieldset>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

fieldset {

width: 400px;

text-align: center;

border-right: none;

border-left: none;

border-bottom: none;

}

legend {

padding-left: 10px;

padding-right: 10px;

}

.btn-create {

margin-top: 20px;

border:1px solid gray;

width: 400px;

text-align: center;

border-radius: 20px;

padding: 10px;

}

</style>

</head>

<body>

<fieldset>

<legend> New to Amazon? </legend>

<div class="btn-create">Create new Account</div>

</fieldset>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

background-color: lightgray;

}

fieldset {

background-color: white;

height: 300px;

border-radius: 50px;

margin-top: 100px;

}

legend {

text-align: center;

width: 80%;

padding: 30px;

height: 40px;

border: 1px solid gray;

background-color: white;

border-radius: 50px;

display: flex;

justify-content: space-around;

box-shadow: 5px 5px 2px black;

}

legend span {

font-size: 50px;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body>

<fieldset>

<legend>

<span class="bi bi-airplane"></span>

<span class="bi bi-train-lightrail-front-fill"></span>

<span class="bi bi-bus-front"></span>

<span class="bi bi-umbrella"></span>

</legend>

</fieldset>

</body>

</html>

**HTML Horizontal Line**

- It is defined using <hr> element.

- It is a horizontal line with size and color.

Syntax:

<hr size="10" noshade color="red" width="50%">

size : thickness of line

noshade : allows to set color for line.

07/04

=====

Details & Summary

Fieldset & Legend

Order and Unordered List

Data List

HR Line

**Text Formatting in HTML**

**1. Font**

- It allows to configure a face, size & color for text.

face : It refers to font family name.

size : It refers to level number 1 to 7.

color : It refers color name, shade or hexadecimal code.

Syntax:

<font face="Arial" size="6" color="#ff00ff"> Your Text </font>

FAQ's:

**1. Why the size is only 7, what it refers to ?**

A. It refers to Level number, which represents default sizes like

small

normal

large

x-large

xx-large

xxx-large etc.

**2. What are web safe fonts?**

A. They refer to default fonts available across all devices.

The web safe fonts are

a) Serif

b) Sans Serif

c) Monospace

Note: Don't use <font> element in modern web design. It is recommended to use

CSS styles.

**2. Text Styles & Effects**

Bold <b>

Strong <strong>

Italics <i>

Emphasized <em>

Underline <u>

Inserted <ins>

Strikeout <strike>

Deleted <del>

Super Script <sup>

Sub Script <sub>

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

Ans: HTML provides different elements for documentation.

The elements are categorized into

a) Design Mode

b) Review Mode

The Review mode elements are

<em>

<strong>

<ins>

<del>

You have to avoid them while going live. [production]

The design mode elements are

<i>

<b>

<u>

<strike>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<div>Font <b>Bold</b> is similar to <strong>Strong</strong> </div>

<div>Font <i>Italics</i> is similar to <em>Emphasized</em> </div>

<div>Text <u>Underline</u> is similar to <ins>Inserted</ins> </div>

<div>Text <strike>Strikeout</strike> is similar to <del>Deleted</del> </div>

<div> 5<sup>th</sup> Anniversary </div>

<div> H<sub>2</sub>O </div>

</body>

</html>

**CSS Text Styles**

font-family : font face

font-size : text size

font-weight : bold

font-style : italics

text-decoration : underline, overline, line-through

[line style, color & size]

Syntax:

h1 {

text-decoration: underline dotted 2px black;

}

Ex:

Flipkart-Mobile

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Flipkart</title>

<style>

body {

display: grid;

grid-template-columns: 3fr 9fr;

padding-top: 30px;

}

.rating {

background-color: green;

color:white;

padding: 5px;

border-radius: 5px;

}

ul {

list-style: none;

margin-left: -40px;

}

li {

line-height: 30px;

color:green;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body>

<div>

</div>

<div>

<div> <font size="5" face="Arial">Apple 16 (White 128 GB)</font> </div>

<br>

<div>

<font face="Arial"><span class="rating">4.6<span class="bi bi-star-fill"></span></span></font>

<font color="gray" face="Arial" size="4"><b> 2,500 ratings & 300 reviews </b></font>

</div>

<br>

<div>

<font size="6" face="Arial"><b>&#8377; 6,242/month</b></font>

<br><br>

<div>

<font face="Arial">12 months No Cost EMI Plan with Bajaj Finserv</font>

<font color="blue" face="Arial">Details</font>

</div>

<br><br>

<font face="Arial" color="green"> <b>Extra &#8377; 5,000 Off</b> </font>

<br><br>

<font face="Arial" size="5"><b> &#8377; 74,900</b> <font color="gray"><strike>&#8377; 79,900/-</strike></font> </font>

</div>

<div>

<h4><font face="Arial">Available Offers</font></h4>

<font face="Arial">

<ul>

<li class="bi bi-tag-fill"> <font color="gray"><b>Bank Offer5%</b> Unlimited Cashback on Flipkart Axis Bank Credit Card</font> </li>

<li class="bi bi-tag-fill"> <font color=gray><b>Bank Offer </b> ₹2500 Off On Flipkart Axis Bank Credit Card Non EMI Transactions</font> </li>

<li class="bi bi-tag-fill"> <font color="gray"><b>Bank Offer</b> ₹4000 Off On Axis Bank Credit Card Transactions.</font> </li>

<li class="bi bi-tag-fill"> <font color="gray"><b>Special Price</b> Get extra ₹5000 off (price inclusive of cashback/coupon)</font> </li>

</ul>

</font>

</div>

</div>

</body>

</html>

**Images in HTML**

**8/04**

====

**Images in HTML**

- Images are of various types.

- Web will not support all types of images.

- The recommended images types for web are:

Type MIME Extension

---------------------------------------------------------------------------------------------------------------------

PNG [ Portable Network Graphics] image/png .png

APNG [ Animated Portable Network Graphics] image/apng .apng

JPEG [Joint Photographic Expert Group] image/jpeg .jpg, .jpeg, .jfif

AVIF [ AV1 Image Format] image/avif .avif, .heif

GIF [ Graphic Interchange Format] image/gif .gif

TIFF [ Tagged Image File Format] image/tiff .tiff

BMP [Bitmap] image/bmp .bmp

ICON [Microsoft Icon] image/icon .ico

SVG [Scalar Vector Graphics] application/xml .svg

WebP [Web Picture] application/webp .webp

**FAQ: What is MIME?**

Ans: Multipurpose Internet Mail Extension

It refers the file content type.

A browser can understand the content type of a file by using its MIME.

Note: Every MIME refers to multiple extensions.

.jpg, .jpeg, .jfif => image/jpeg

**PNG & APNG**

- High Definition

- High Pixel Depth

- More Colors

- Requires more memory

- APNG can be animated

- Always use PNG for downloadable images.

- Don't use them for on screen presentation.

**JPEG**

- High Definition

- Less pixel depth when compared to PNG

- More colors

- It is compressed image format.

- Requires less memory

- It is good for on-screen presentation.

**AVIF**

- Similar to JPG

- 50% more compressed than JPG

- They not supported across all browsers.

- They compatibility issues.

- They use HEIF. [High Efficiency Image Format]

- Less memory and fast in loading.

**GIF**

- Less colors

- It supports only 256 colors.

- Size is same as its original format.

- They are good for logos, bullets, buttons, patterns, borders etc.

- It can be animated.

**TIFF**

- It is a sliced image format.

- It is high definition & quality

- It uses more memory.

- It loads sliced image portion on web page.

- It is good for scanned documents.

**SVG**

- Scalar Vector Graphics

- Vector Graphics is not a pixel based image.

- It uses XML as language.

- It is good for Maps, Architectural diagrams, icons etc.

**Bitmap**

- It is a binary image format.

- It is good for storing in any database or data store.

- It is also good in transporting via network.

- However it require COM to Marshal [Common Object Model] the content.

- It is blocked by firewalls.

**WebP**

- It is embedded image into page.

- It will not allow to separate image from page.

- It a secured image in page.

**Embed Image in Page:**

1. You can use <img> element to embed images.

2. To keep image SEO friendly you can enclose image in <Figure> element with

<Figcaption>

Syntax:

<figure>

<img>

<figcaption> Title </figcaption>

</figure>

**Image Attributes:**

**1. src** : It refers to the image name and path.

**2. width & height** : sets height and width for image. You can use pixels or %.

Size in % makes image fluid.

**9/04**

====

**Image Attributes**

**1. src**

**2. width & height**

**3. alt** : It defines the alternative text to display when image fails to load.

**4. title** : It defines the text to display when mouse is over image.

Syntax:

<img src="./images/pic.jpg" alt="alternative text" title="screen tips">

**5. border** : It sets border for image by specified pixels.

**6. align** : It aligns image left or right. It is used for wrapping text around image

**7. hspace** : It sets horizontal padding between text and image

**8. vspace** : It set vertical padding between text and image

Syntax:

<img src="./images/pic.jpg" align="left" hspace="20" vspace="20">

**9. crossorigin** : It refers to CORS [Cross Origin Resource Sharing].

It can restrict accessibility of image by using various authentication

Techniques.

a) Anonymous

b) use-credentials

Syntax:

<img src="some\_path" crossorigin="use-credentials">

**10. decoding** : It defines how image can load along with other content in page.

a) sync

b) async

c) auto

Syntax:

<img src="path" decoding="async | sync | auto">

**11. importance** : It sets priority for image. You can load according to priority.

a) low

b) high

c) auto

Syntax:

<img src="path" importance="low | high | auto">

**12. loading** : It defines the loading technique for image, which can be

a) eager

b) lazy

Lazy loading is a technique of loading only when required.

Syntax:

<img src="path" loading="lazy | eager">

**13. srcset** : It configure a set of images for element.

It requires styles to rotate images according to state and situation.

Syntax:

<img srcset="collection\_of\_images">

Ex: Cards Design - Naresh IT cards

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Naresh IT</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

.card {

width: 250px;

margin: 20px;

border-radius: 10px;

border:1px solid gray;

box-shadow: 2px 2px 2px black;

font-family: Arial;

}

.card-img {

width: 100%;

border-top-left-radius: 10px;

border-top-right-radius: 10px;

height: 150px;

}

.card-header {

height: 150px;

}

.card-body {

padding: 20px;

}

.card-title {

text-align: center;

font-size: 22px;

font-weight: bold;

}

.rating {

text-align: center;

color:goldenrod;

padding-top: 20px;

}

ul {

list-style: none;

margin-left: -40px;

}

li {

line-height: 30px;

}

.card-footer {

display: grid;

grid-template-columns: 6fr 6fr;

text-align: center;

}

.btn-enroll {

background-color: blue;

color:white;

padding: 10px;

border-bottom-left-radius: 10px;

}

.btn-share {

background-color: whitesmoke;

padding: 10px;

border-bottom-right-radius: 10px;

}

section {

display: flex;

flex-direction: row;

}

</style>

</head>

<body>

<section>

<div class="card">

<div class="card-header">

<img src="./images/power-bi.png" class="card-img">

</div>

<div class="card-body">

<div>

<div class="card-title">Power PI</div>

<div class="rating">

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

</div>

</div>

<div>

<ul>

<li class="bi bi-calendar-date"> <span>Start Date</span> <span>:</span> <span>10 Apr 2025</span> </li>

<li class="bi bi-person"> <span>By</span> <span>:</span> <span>Mr.Someone</span> </li>

<li class="bi bi-clock"> <span>Duration</span> <span>:</span> <span> 60 Days </span> </li>

</ul>

</div>

</div>

<div class="card-footer">

<div class="btn-enroll">

Enroll

</div>

<div class="btn-share">

<span class="bi bi-share"></span> Share

</div>

</div>

</div>

<div class="card">

<div class="card-header">

<img src="./images/python.png" class="card-img">

</div>

<div class="card-body">

<div>

<div class="card-title">Python</div>

<div class="rating">

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

</div>

</div>

<div>

<ul>

<li class="bi bi-calendar-date"> <span>Start Date</span> <span>:</span> <span>10 Apr 2025</span> </li>

<li class="bi bi-person"> <span>By</span> <span>:</span> <span>Mr.Someone</span> </li>

<li class="bi bi-clock"> <span>Duration</span> <span>:</span> <span> 60 Days </span> </li>

</ul>

</div>

</div>

<div class="card-footer">

<div class="btn-enroll">

Enroll

</div>

<div class="btn-share">

<span class="bi bi-share"></span> Share

</div>

</div>

</div>

<div class="card">

<div class="card-header">

<img src="./images/react.png" class="card-img">

</div>

<div class="card-body">

<div>

<div class="card-title">React JS</div>

<div class="rating">

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

</div>

</div>

<div>

<ul>

<li class="bi bi-calendar-date"> <span>Start Date</span> <span>:</span> <span>10 Apr 2025</span> </li>

<li class="bi bi-person"> <span>By</span> <span>:</span> <span>Mr.Someone</span> </li>

<li class="bi bi-clock"> <span>Duration</span> <span>:</span> <span> 60 Days </span> </li>

</ul>

</div>

</div>

<div class="card-footer">

<div class="btn-enroll">

Enroll

</div>

<div class="btn-share">

<span class="bi bi-share"></span> Share

</div>

</div>

</div>

<div class="card">

<div class="card-header">

<img src="./images/data sci.png" class="card-img">

</div>

<div class="card-body">

<div>

<div class="card-title">Data Science</div>

<div class="rating">

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

</div>

</div>

<div>

<ul>

<li class="bi bi-calendar-date"> <span>Start Date</span> <span>:</span> <span>10 Apr 2025</span> </li>

<li class="bi bi-person"> <span>By</span> <span>:</span> <span>Mr.Someone</span> </li>

<li class="bi bi-clock"> <span>Duration</span> <span>:</span> <span> 60 Days </span> </li>

</ul>

</div>

</div>

<div class="card-footer">

<div class="btn-enroll">

Enroll

</div>

<div class="btn-share">

<span class="bi bi-share"></span> Share

</div>

</div>

</div>

</section>

</body>

</html>

Ex: Podcast cards

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

.card {

width: 400px;

border:1px solid gray;

box-shadow: 3px 3px 2px black;

border-radius: 5px;

margin: 20px;

display: grid;

grid-template-columns: 6fr 6fr;

height: 200px;

}

.card-img {

height: 200px;

border-top-right-radius: 5px;

border-bottom-right-radius: 5px;

}

.card-header {

text-align: center;

padding: 20px;

font-family: Arial;

}

.card-title {

font-size: 30px;

padding-top: 10px;

padding-bottom: 30px;

}

.bi-chevron-bar-left {

font-size: 25px;

}

.bi-chevron-bar-right {

font-size: 25px;

}

.bi-play-fill {

font-size: 50px;

}

.controls {

display: flex;

justify-content: center;

align-items: center;

}

</style>

</head>

<body>

<div class="card">

<div class="card-header">

<div class="card-title">Arjit Singh</div>

<div class="controls">

<span class="bi bi-chevron-bar-left"></span>

<span class="bi bi-play-fill"></span>

<span class="bi bi-chevron-bar-right"></span>

</div>

</div>

<div class="card-body">

<img src="./images/arjitsingh.jpg" class="card-img">

</div>

</div>

<div class="card">

<div class="card-header">

<div class="card-title">Arjit Singh</div>

<div class="controls">

<span class="bi bi-chevron-bar-left"></span>

<span class="bi bi-play-fill"></span>

<span class="bi bi-chevron-bar-right"></span>

</div>

</div>

<div class="card-body">

<img src="./images/arjitsingh.jpg" class="card-img">

</div>

</div>

</body>

</html>

10/04

====

Image Attributes

**1. src**

**2. width & height**

**3. alt** : It defines the alternative text to display when image fails to load.

**4. title** : It defines the text to display when mouse is over image.

Syntax:

<img src="./images/pic.jpg" alt="alternative text" title="screen tips">

**5. border** : It sets border for image by specified pixels.

**6. align** : It aligns image left or right. It is used for wrapping text around image

**7. hspace** : It sets horizontal padding between text and image

**8. vspace** : It set vertical padding between text and image

Syntax:

<img src="./images/pic.jpg" align="left" hspace="20" vspace="20">

**9. crossorigin** : It refers to CORS [Cross Origin Resource Sharing].

It can restrict accessibility of image by using various authentication

techniques.

a) anonymous

b) use-credentials

Syntax:

<img src="some\_path" crossorigin="use-credentials">

**10. decoding** : It defines how image can load along with other content in page.

a) sync

b) async

c) auto

Syntax:

<img src="path" decoding="async | sync | auto">

**11. importance** : It sets priority for image. You can load according to priority.

a) low

b) high

c) auto

Syntax:

<img src="path" importance="low | high | auto">

**12. loading** : It defines the loading technique for image, which can be

a) eager

b) lazy

Lazy loading is a technique of loading only when required.

Syntax:

<img src="path" loading="lazy | eager">

**13. srcset** : It configure a set of images for element.

It requires styles to rotate images according to state and situation.

Syntax:

<img srcset="collection\_of\_images">

Ex: Cards Design - Naresh IT cards

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Naresh IT</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

.card {

width: 250px;

margin: 20px;

border-radius: 10px;

border:1px solid gray;

box-shadow: 2px 2px 2px black;

font-family: Arial;

}

.card-img {

width: 100%;

border-top-left-radius: 10px;

border-top-right-radius: 10px;

height: 150px;

}

.card-header {

height: 150px;

}

.card-body {

padding: 20px;

}

.card-title {

text-align: center;

font-size: 22px;

font-weight: bold;

}

.rating {

text-align: center;

color:goldenrod;

padding-top: 20px;

}

ul {

list-style: none;

margin-left: -40px;

}

li {

line-height: 30px;

}

.card-footer {

display: grid;

grid-template-columns: 6fr 6fr;

text-align: center;

}

.btn-enroll {

background-color: blue;

color:white;

padding: 10px;

border-bottom-left-radius: 10px;

}

.btn-share {

background-color: whitesmoke;

padding: 10px;

border-bottom-right-radius: 10px;

}

section {

display: flex;

flex-direction: row;

}

</style>

</head>

<body>

<section>

<div class="card">

<div class="card-header">

<img src="./images/power-bi.png" class="card-img">

</div>

<div class="card-body">

<div>

<div class="card-title">Power PI</div>

<div class="rating">

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

</div>

</div>

<div>

<ul>

<li class="bi bi-calendar-date"> <span>Start Date</span> <span>:</span> <span>10 Apr 2025</span> </li>

<li class="bi bi-person"> <span>By</span> <span>:</span> <span>Mr.Someone</span> </li>

<li class="bi bi-clock"> <span>Duration</span> <span>:</span> <span> 60 Days </span> </li>

</ul>

</div>

</div>

<div class="card-footer">

<div class="btn-enroll">

Enroll

</div>

<div class="btn-share">

<span class="bi bi-share"></span> Share

</div>

</div>

</div>

<div class="card">

<div class="card-header">

<img src="./images/python.png" class="card-img">

</div>

<div class="card-body">

<div>

<div class="card-title">Python</div>

<div class="rating">

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

</div>

</div>

<div>

<ul>

<li class="bi bi-calendar-date"> <span>Start Date</span> <span>:</span> <span>10 Apr 2025</span> </li>

<li class="bi bi-person"> <span>By</span> <span>:</span> <span>Mr.Someone</span> </li>

<li class="bi bi-clock"> <span>Duration</span> <span>:</span> <span> 60 Days </span> </li>

</ul>

</div>

</div>

<div class="card-footer">

<div class="btn-enroll">

Enroll

</div>

<div class="btn-share">

<span class="bi bi-share"></span> Share

</div>

</div>

</div>

<div class="card">

<div class="card-header">

<img src="./images/react.png" class="card-img">

</div>

<div class="card-body">

<div>

<div class="card-title">React JS</div>

<div class="rating">

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

</div>

</div>

<div>

<ul>

<li class="bi bi-calendar-date"> <span>Start Date</span> <span>:</span> <span>10 Apr 2025</span> </li>

<li class="bi bi-person"> <span>By</span> <span>:</span> <span>Mr.Someone</span> </li>

<li class="bi bi-clock"> <span>Duration</span> <span>:</span> <span> 60 Days </span> </li>

</ul>

</div>

</div>

<div class="card-footer">

<div class="btn-enroll">

Enroll

</div>

<div class="btn-share">

<span class="bi bi-share"></span> Share

</div>

</div>

</div>

<div class="card">

<div class="card-header">

<img src="./images/data sci.png" class="card-img">

</div>

<div class="card-body">

<div>

<div class="card-title">Data Science</div>

<div class="rating">

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

<span class="bi bi-star-fill"></span>

</div>

</div>

<div>

<ul>

<li class="bi bi-calendar-date"> <span>Start Date</span> <span>:</span> <span>10 Apr 2025</span> </li>

<li class="bi bi-person"> <span>By</span> <span>:</span> <span>Mr.Someone</span> </li>

<li class="bi bi-clock"> <span>Duration</span> <span>:</span> <span> 60 Days </span> </li>

</ul>

</div>

</div>

<div class="card-footer">

<div class="btn-enroll">

Enroll

</div>

<div class="btn-share">

<span class="bi bi-share"></span> Share

</div>

</div>

</div>

</section>

</body>

</html>

Ex: Podcast cards

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

.card {

width: 400px;

border:1px solid gray;

box-shadow: 3px 3px 2px black;

border-radius: 5px;

margin: 20px;

display: grid;

grid-template-columns: 6fr 6fr;

height: 200px;

}

.card-img {

height: 200px;

border-top-right-radius: 5px;

border-bottom-right-radius: 5px;

}

.card-header {

text-align: center;

padding: 20px;

font-family: Arial;

}

.card-title {

font-size: 30px;

padding-top: 10px;

padding-bottom: 30px;

}

.bi-chevron-bar-left {

font-size: 25px;

}

.bi-chevron-bar-right {

font-size: 25px;

}

.bi-play-fill {

font-size: 50px;

}

.controls {

display: flex;

justify-content: center;

align-items: center;

}

</style>

</head>

<body>

<div class="card">

<div class="card-header">

<div class="card-title">Arjit Singh</div>

<div class="controls">

<span class="bi bi-chevron-bar-left"></span>

<span class="bi bi-play-fill"></span>

<span class="bi bi-chevron-bar-right"></span>

</div>

</div>

<div class="card-body">

<img src="./images/arjitsingh.jpg" class="card-img">

</div>

</div>

<div class="card">

<div class="card-header">

<div class="card-title">Arjit Singh</div>

<div class="controls">

<span class="bi bi-chevron-bar-left"></span>

<span class="bi bi-play-fill"></span>

<span class="bi bi-chevron-bar-right"></span>

</div>

</div>

<div class="card-body">

<img src="./images/arjitsingh.jpg" class="card-img">

</div>

</div>

</body>

</html>

**10/04**

====

Ex: Amazon Cards

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Amazon</title>

<style>

.content-row {

display: grid;

grid-template-columns: 3fr 3fr 3fr 3fr;

padding: 10px;

margin-bottom: 30px;

}

.card {

padding: 10px;

margin: 5px;

background-color: white;

box-shadow: 2px 2px 2px gray;

height: 400px;

font-family: Arial;

}

section {

margin-top: -250px;

}

.card-title {

font-size: 22px;

font-weight: bold;

}

.row {

display: grid;

grid-template-columns: 6fr 6fr;

grid-gap: 5px;

}

.card-img {

width: 100%;

}

.card-header {

height: 70px;

}

.card-footer {

text-align: center;

padding-top: 20px;

}

.more {

color:blue;

}

</style>

</head>

<body>

<header>

<img src="./images/amazon-banner.png" width="100%">

</header>

<section>

<div class="content-row">

<div class="card">

<div class="card-header">

<div class="card-title">Appliances for your home | Up to 55% off</div>

</div>

<div class="card-body">

<div class="row">

<div><img class="card-img" src="./images/a1.jpg">

<p>Air Conditioners</p>

</div>

<div><img class="card-img" src="./images/a2.jpg">

<p>Refrigerators</p>

</div>

</div>

<div class="row">

<div><img class="card-img" src="./images/a3.jpg"> <p>Microwaves</p> </div>

<div><img class="card-img" src="./images/a4.jpg"> <p>Washing Machines</p> </div>

</div>

</div>

<div class="card-footer">

<div class="more">see more</div>

</div>

</div>

<div class="card">

<div class="card-header">

<div class="card-title">Revamp your home in style</div>

</div>

<div class="card-body">

<div class="row">

<div><img class="card-img" src="./images/a5.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a6.jpg"> <p>Microwaves</p></div>

</div>

<div class="row">

<div><img class="card-img" src="./images/a7.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a8.jpg"> <p>Microwaves</p></div>

</div>

</div>

<div class="card-footer">

<div class="more">see all</div>

</div>

</div>

<div class="card">

<div class="card-header">

<div class="card-title"> Starting ₹149 | Headphones</div>

</div>

<div class="card-body">

<div class="row">

<div><img class="card-img" src="./images/a9.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a10.jpg"> <p>Microwaves</p></div>

</div>

<div class="row">

<div><img class="card-img" src="./images/a11.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a12.jpg"> <p>Microwaves</p></div>

</div>

</div>

<div class="card-footer">

<div class="more">explore all</div>

</div>

</div>

<div class="card">

<div class="card-header">

<div class="card-title">Automotive essentials | Up to 60% off</div>

</div>

<div class="card-body">

<div class="row">

<div><img class="card-img" src="./images/a1.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a2.jpg"> <p>Microwaves</p></div>

</div>

<div class="row">

<div><img class="card-img" src="./images/a3.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a4.jpg"> <p>Microwaves</p></div>

</div>

</div>

<div class="card-footer">

<div class="more">see more</div>

</div>

</div>

</div>

<div class="content-row">

<div class="card">

<div class="card-header">

<div class="card-title">Appliances for your home | Up to 55% off</div>

</div>

<div class="card-body">

<div class="row">

<div><img class="card-img" src="./images/a1.jpg">

<p>Air Conditioners</p>

</div>

<div><img class="card-img" src="./images/a2.jpg">

<p>Refrigerators</p>

</div>

</div>

<div class="row">

<div><img class="card-img" src="./images/a3.jpg"> <p>Microwaves</p> </div>

<div><img class="card-img" src="./images/a4.jpg"> <p>Washing Machines</p> </div>

</div>

</div>

<div class="card-footer">

<div class="more">see more</div>

</div>

</div>

<div class="card">

<div class="card-header">

<div class="card-title">Revamp your home in style</div>

</div>

<div class="card-body">

<div class="row">

<div><img class="card-img" src="./images/a5.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a6.jpg"> <p>Microwaves</p></div>

</div>

<div class="row">

<div><img class="card-img" src="./images/a7.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a8.jpg"> <p>Microwaves</p></div>

</div>

</div>

<div class="card-footer">

<div class="more">see all</div>

</div>

</div>

<div class="card">

<div class="card-header">

<div class="card-title"> Starting ₹149 | Headphones</div>

</div>

<div class="card-body">

<div class="row">

<div><img class="card-img" src="./images/a9.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a10.jpg"> <p>Microwaves</p></div>

</div>

<div class="row">

<div><img class="card-img" src="./images/a11.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a12.jpg"> <p>Microwaves</p></div>

</div>

</div>

<div class="card-footer">

<div class="more">explore all</div>

</div>

</div>

<div class="card">

<div class="card-header">

<div class="card-title">Automotive essentials | Up to 60% off</div>

</div>

<div class="card-body">

<div class="row">

<div><img class="card-img" src="./images/a1.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a2.jpg"> <p>Microwaves</p></div>

</div>

<div class="row">

<div><img class="card-img" src="./images/a3.jpg"> <p>Microwaves</p></div>

<div><img class="card-img" src="./images/a4.jpg"> <p>Microwaves</p></div>

</div>

</div>

<div class="card-footer">

<div class="more">see more</div>

</div>

</div>

</div>

</section>

</body>

</html>

**Hyperlinks in HTML**

- Every web application must provide a simplified navigation mechanism.

- User must able to reach to the required location with single click.

- Websites or applications use a "Site Map" for navigation among all the services provided by the business.

- Navigation is designed using Hyperlink.

- Hyperlink is a clickable text, picture or graphic that navigates user from one location to another when clicked.

- HTML can create hyperlinks by using "<a>" anchor element.

- Hyperlinks are classified into 2 types

a) Intra document links

b) Inter document links

**Intra Document Links:**

- Intra document link allow navigation from one location to another within page.

- The target location is marked with an "ID".

<footer id="footer">

<aside id="ads">

<aside id="social">

- You can reach to the marked location by requesting ID with "#" hash reference.

- You can request from URL in address bar or from a hyperlink.

http://server.com/page.html#ads

<a href="#footer"> Text | Image | Graphics </a>

<a href="#social"> any content </a>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Intra Links</title>

<style>

header {

background-color: black;

color:white;

padding: 2px;

text-align: center;

}

section {

display: grid;

grid-template-columns: 2fr 10fr;

margin-top: 20px;

}

main {

height: 500px;

overflow: auto;

}

.menu {

list-style: none;

margin-left: -30px;

font-family: Arial;

font-size: 20px;

}

li {

margin-top: 10px;

margin-bottom: 10px;

width: 100px;

background-color: black;

color:white;

padding: 10px;

border-radius: 5px;

}

nav a {

color:white;

text-decoration: none;

}

li:hover {

background-color: blue;

}

main a:visited {

color:green;

}

</style>

</head>

<body>

<header>

<h3>Shopper</h3>

</header>

<section>

<nav>

<ul class="menu">

<li><a href="#home">Home</a></li>

<li><a href="#kids">Kids</a></li>

<li><a href="#women">Women</a></li>

<li><a href="#men">Men</a></li>

</ul>

</nav>

<main>

<h3 id="home">Home</h3>

<p>Lorem, ipsum dolor <span>Offers on <a href="#iphone">iPhone</a></span> sit amet consectetur adipisicing elit. Porro omnis dicta blanditiis, ea quo beatae magnam! Fugiat ullam, assumenda tenetur corrupti incidunt temporibus voluptatum odit. Similique quis reprehenderit quas corporis. Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati sint repellat iste, a voluptates distinctio ducimus nisi est fugiat. Perspiciatis, maxime. Voluptate officiis esse quaerat animi soluta voluptates autem quae.</p>

<h3 id="women">Women Shopping</h3>

<img src="./images/women-fashion.jpg" width="200" height="200">

<div>

<a href="#home">Back to Top</a>

</div>

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Porro omnis dicta blanditiis, ea quo beatae magnam! Fugiat ullam, assumenda tenetur corrupti incidunt temporibus voluptatum odit. Similique quis reprehenderit quas corporis. Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati sint repellat iste, a voluptates distinctio ducimus nisi est fugiat. Perspiciatis, maxime. Voluptate officiis esse quaerat animi soluta voluptates autem quae.</p>

<h3 id="men">Men Shopping</h3>

<img src="./images/men-fashion.jpg" width="200" height="200">

<div>

<a href="#home">Back to Top</a>

</div>

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Porro omnis dicta blanditiis, ea quo beatae magnam! Fugiat ullam, assumenda tenetur corrupti incidunt temporibus voluptatum odit. Similique quis reprehenderit quas corporis. Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati sint repellat iste, a voluptates distinctio ducimus nisi est fugiat. Perspiciatis, maxime. Voluptate officiis esse quaerat animi soluta voluptates autem quae.</p>

<figure id="iphone">

<img src="./images/iphone-white.jpg" width="200" height="200">

<figcaption>iPhone 16 (White) 128 GB</figcaption>

</figure>

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Porro omnis dicta blanditiis, ea quo beatae magnam! Fugiat ullam, assumenda tenetur corrupti incidunt temporibus voluptatum odit. Similique quis reprehenderit quas corporis. Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati sint repellat iste, a voluptates distinctio ducimus nisi est fugiat. Perspiciatis, maxime. Voluptate officiis esse quaerat animi soluta voluptates autem quae.</p>

<h3 id="kids">Kids Shopping</h3>

<img src="./images/kids-fashion.jpg" width="200" height="200">

<div>

<a href="#home">Back to Top</a>

</div>

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Porro omnis dicta blanditiis, ea quo beatae magnam! Fugiat ullam, assumenda tenetur corrupti incidunt temporibus voluptatum odit. Similique quis reprehenderit quas corporis. Lorem ipsum dolor sit amet consectetur adipisicing elit. Obcaecati sint repellat iste, a voluptates distinctio ducimus nisi est fugiat. Perspiciatis, maxime. Voluptate officiis esse quaerat animi soluta voluptates autem quae.</p>

</main>

</section>

</body>

</html>

FAQ's:

**1. How to change the color for visited and active links?**

A. By using <body> element attributes

- alink

- valink

By using CSS classes

:active

:visited

Syntax:

<body alink="red" vlink="green">

a:visited {

color:green;

}

a:active

color:red;

}

**2. How to remove underline for hyperlink?**

A. By configure "text-decoration" set to "none".

Syntax:

a {

text-decoration: none;

}

3. What is difference between "src" & "href" ?

A. "src" is a getter.

"href" is a setter. It sets a value into the browser address bar.

**4. If you don't have much content to scroll to the marked location, how to highlight**

the current content?

A. By using CSS ":target" class you can highlight the current context.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.card {

width: 200px;

padding: 20px;

box-shadow: 4px 4px 2px black;

border:1px solid black;

margin: 20px;

}

.card:target {

background-color: black;

color:white;

border-radius: 100%;

height: 200px;

text-align: center;

}

section {

margin-top: 50px;

display: flex;

flex-direction: row;

}

nav {

display: flex;

justify-content: center;

padding: 20px;

}

nav span {

font-size: 25px;

padding-left: 20px;

padding-right: 20px;

}

</style>

</head>

<body>

<nav>

<span><a href="#html">HTML</a></span>

<span><a href="#css">CSS</a></span>

<span><a href="#js">JavaScript</a></span>

</nav>

<section>

<div class="card" id="html">

<h3>HTML</h3>

<p>It is a markup language.</p>

</div>

<div class="card" id="css">

<h3>CSS</h3>

<p>It is used to style DOM.</p>

</div>

<div class="card" id="js">

<h3>JavaScript</h3>

<p>It is used to manipulate DOM.</p>

</div>

</section>

</body>

</html>

**11/04**

=====

**Inter Documents Links**

- Inter document links handle various interactions.

- It allows

a) Navigation to a file

b) Navigation to any URL

c) Access to any browser supported application

d) Access to any client side function

Navigate to file:

- You can create a link for any existing file in your project.

- You can link to any type of file.

- Every linked document can't open in browser.

- Browser requires relative plugin's to open the file content.

- If plugin is not supported then the linked file will download.

Syntax:

<a href="./shopper.html"> Shopper Template </a>

<a href="./images/pic.jpg"> Photo </a>

<a href="./docs/catalog.pdf"> Catalog </a>

**FAQ: How to configure a link that always downloads the file?**

Ans: By using <a> anchor "download" attribute.

Syntax:

<a href="./images/pic.jpg" download="any\_file\_name">

<a href="./images/pic.jpg" download>

Navigate to URL:

- You can configure a hyperlink that navigate user from current page to any remote resource location.

- It can be a webpage, image, or any document.

- URL must have proper

a) protocol

b) domain

c) reference path for resource [optional]

Syntax:

<a href="https://www.amazon.in"&gt; Amazon Shopping </a>

<a href="http://www.server.com/path/file.jpg"&gt; Remote Photo </a>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<title>Document</title>

</head>

<body>

<ol>

<li> <a href="./shopper-template.html">Shopper Template</a> </li>

<li> <a href="./images/women-fashion.jpg"> Women Fashion </a></li>

<li> <a href="./images/kids-fashion.jpg" download="DSC00153.jpg"> <img src="./images/kids-fashion.jpg" width="100" height="100"> <span class="bi bi-download"></span> </a></li>

<li> <a href="./docs/catalog.pdf"> Fitness Catalog </a> </li>

<li> <a href="./docs/ishop.docx"> Shopping Project Document </a> </li>

<li> <a href="https://www.amazon.in"&gt; Amazon India </a></li>

<li> <a href="https://www.amazon.com"&gt; Amazon US </a></li>

<li> <a href="http://mars.jpl.nasa.gov/msl-raw-images/proj/msl/redops/ods/surface/sol/01000/opgs/edr/fcam/FLB\_486265257EDR\_F0481570FHAZ00323M\_.JPG"&gt; Mars Rover Photo</a></li>

</ol>

</body>

</html>

**Accessing browser supported application:**

- Hyperlink can invoke various applications from your browser.

- Applications must support browser, the popular apps are

a) Email App "mailto"

b) Phone App "tel"

c) Skype App "skype"

d) WhatsApp "wa.me"

Syntax:

<a href="youremail@gmail.com"> Mail </a>

<a href="tel:+919999112234"> Call </a>

<a hre="https://wa.me/your\_number"&gt; WhatsApp </a>

**Accessing Client Side Function:**

- Client side functions are written using client side script like "JavaScript".

- These functions are used in BOM and DOM interactions.

**Browser Object Model**

- window

- location

- navigator

- history

- document

**Document Object Model**

- Event Binding

- Style Binding

- Class Binding

- Data Binding etc.

- You can invoke script functions by using "javascript" reference.

Syntax:

<a href="javascript:function(){}"> </a>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<title>Document</title>

</head>

<body>

<ol>

<li> For more details you can <a href="info@nareshit.in"><span class="bi bi-envelope"> info@nareshit.in</span> </a></li>

<li> You can give a missed call to <a href="tel:+919876543211"> +919876543211 </a> </li>

<li> You message on <span class="bi bi-whatsapp"> <a href="https://wa.me/+919876543211"&gt; +919876543211 </a> </span> </li>

<li> Skype Us <span class="bi bi-skype"> <a href="skype:youraccount?call"> Nareshit@outlook.com </a> </span> </li>

<li> <span class="bi bi-printer-fill"> <a href="javascript:window.print()">Print Page </a> </span> </li>

</ol>

</body>

</html>

**FAQ's:**

**1. How to open linked document in a new tab?**

A. By using <a> anchor "target" attribute set to "\_blank".

Syntax:

<a href="./images/fashion.jpg" target="\_blank"> Fashion </a>

**2. How to open linked document in a new window?**

A. By using JavaScript "window.open()" method.

Syntax:

<a href="javascript:window.open('path', 'title', ' features\_for\_window' )">

window features are : width, height, buttons etc.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<title>Document</title>

</head>

<body>

<ol>

<li> <a href="./images/women-fashion.jpg"> Women Fashion </a> </li>

<li> <a href="./images/kids-fashion.jpg" target="\_blank"> Kids Fashion </a> </li>

<li> <a href="javascript:window.open('./images/men-fashion.jpg','Men','width=300 height=400')"> Men Fashion </a> </li>

<li> <a href="./docs/catalog.pdf" target="\_blank"> Catalog </a></li>

</ol>

</body>

</html>

**3. How to open linked document in the same window along with existing content in page?**

A: By using HTML 5 <iframe> element.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<title>Document</title>

<style>

body {

display: grid;

grid-template-columns: 2fr 10fr;

}

iframe {

border:none;

}

</style>

</head>

<body>

<nav>

<ol>

<li> <a href="./images/women-fashion.jpg"> Women Fashion </a> </li>

<li> <a href="./images/kids-fashion.jpg" target="\_blank"> Kids Fashion </a> </li>

<li> <a href="javascript:window.open('./images/men-fashion.jpg','Men','width=300 height=400')"> Men Fashion </a> </li>

<li> <a href="./docs/catalog.pdf" target="main-frame"> Catalog </a></li>

<li> <a href="./images/a9.jpg" target="main-frame"> <img src="./images/a9.jpg" width="50" height="50"> </a> </li>

<li> <a href="./shopper-template.html" target="main-frame">Shopper</a> </li>

<li> <a href="https://www.youtube.com/embed/pLkzLBCBeuA&quot; target="main-frame"> Java Functions </a> </li>

<li> <a href="https://www.youtube.com/embed/OzltVfNBKt4&quot; target="main-frame"> AWS Workshop </a> </li>

</ol>

</nav>

<main>

<iframe name="main-frame" width="100%" height="550"></iframe>

</main>

</body>

</html>

**15/04**

**Tables in HTML**

- Tables are used to design Grid components.

- Grid is a collection of rows and columns.

- Data Grid is a component that is used for presenting and manipulating data.

- HTML table elements are

<table>

<caption>

<thead>

<tbody>

<tfoot>

<tr>

<th>

<td>

<colgroup>

**Table Attributes:**

**1. Frame, Rules & Border**

**frame** : It sets a frame for table. It have values: box, void, above, below, rhs, lhs.

<table frame="box | void..">

It can take only one value for attribute. Hence you can't apply multiple.

**rules** : It sets rules [lines] for rows, columns and groups.

It have values : all, rows, cols, groups, none.

<table rules="all | rows | cols.. ">

**border**: It sets border for cells. It can have values 0=false, 1=true.

<table border="1">

You can't display border if rules attribute is defined for table.

Make sure that rules attribute is removed from table.

**2. Cell Space and Padding**

**cellspacing** : It sets space between cells.

**cellpadding** : It sets space between border and content inside cell.

Syntax:

<table cellspacing="5" cellpadding="5" border="1">

**3. Align Horizontal & Vertical**

**align** : It aligns content left, center, right or justified horizontally.

**valign** : It aligns content top, center or bottom vertically.

Syntax:

<table align="center">

<tbody align="center">

<tr align="center">

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

**4. Back color & Image**

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

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

Syntax:

<table bgcolor="yellow">

<tr background="./images/pic.jpg">

**5. Width & Height**

**width** : It is defined only for table, td or th.

It sets to entire column.

**height** : It is defined only for tr & td.

It sets to entire row.

Syntax:

<th width="20">

<tr height="30">

**6. Merging rows & columns**

**colspan** : It merges specified number of cells into one cell.

It is used to merge all columns into one column.

**rowspan** : it is used to merge all rows into one row.

Note: You can apply only for td or th.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<table border="1" width="100%">

<thead>

<tr>

<th colspan="2">Name</th>

<th colspan="3">Address</th>

</tr>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>City</th>

<th>State</th>

<th>Postal Code</th>

</tr>

</thead>

<tbody>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td rowspan="7" align="center"> D<br>E<br>L<br>H<br>I </td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="5" align="center">&copy; copyright 2025</td>

</tr>

</tfoot>

</table>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Table</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

table {

font-family: Arial;

}

caption {

text-align: left;

margin-bottom: 10px;

font-weight: bold;

color:gray;

}

.img-team {

width: 20px;

padding-right: 10px;

}

.won {

display: inline-block;

width: 15px;

height: 15px;

border-radius: 15px;

border: 1px solid green;

padding: 5px;

text-align: center;

font-size: 14px;

color:green;

}

.lost {

display: inline-block;

width: 15px;

height: 15px;

border-radius: 15px;

border: 1px solid red;

padding: 5px;

text-align: center;

font-size: 14px;

color:red;

}

</style>

</head>

<body>

<table width="100%" border="0" rules="rows" frame="void" cellspacing="5" cellpadding="10">

<caption>IPL Points Table</caption>

<colgroup span="11"></colgroup>

<thead>

<tr>

<th colspan="12">IPL - 2025</th>

</tr>

<tr >

<th width="20">POS</th>

<th>&nbsp;</th>

<th>TEAM</th>

<th>P</th>

<th>W</th>

<th>L</th>

<th>NR</th>

<th>NRR</th>

<th>FOR</th>

<th>AGAINST</th>

<th>PTS</th>

<th width="150">RECENT FORM</th>

</tr>

</thead>

<tbody >

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

<td>1</td>

<td><span class="bi bi-dash"></span></td>

<td> <img src="./images/GT.png" align="left" class="img-team"> <span>GT</span> </td>

<td>6</td>

<td>4</td>

<td>2</td>

<td>0</td>

<td>1.081</td>

<td>1148/114.3</td>

<td>1069/119.3</td>

<td>8</td>

<td align="center">

<span class="lost">L</span>

<span class="won">W</span>

<span class="won">W</span>

<span class="won">W</span>

<span class="won">W</span>

</td>

</tr>

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

<td>2</td>

<td><span class="bi bi-dash"></span></td>

<td> <img src="./images/DC.png" align="left" class="img-team"> <span>DC</span> </td>

<td>5</td>

<td>4</td>

<td>2</td>

<td>0</td>

<td>1.081</td>

<td>1148/114.3</td>

<td>1069/119.3</td>

<td>8</td>

<td align="center">

<span class="lost">L</span>

<span class="won">W</span>

<span class="won">W</span>

<span class="won">W</span>

<span class="won">W</span>

</td>

</tr>

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

<td>3</td>

<td><span class="bi bi-dash"></span></td>

<td> <img src="./images/RCB.png" align="left" class="img-team"> <span>RCB</span> </td>

<td>6</td>

<td>4</td>

<td>2</td>

<td>0</td>

<td>1.081</td>

<td>1148/114.3</td>

<td>1069/119.3</td>

<td>8</td>

<td align="center">

<span class="lost">L</span>

<span class="won">W</span>

<span class="lost">L</span>

<span class="won">W</span>

<span class="lost">L</span>

</td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="12">

<div>W - Won</div>

<div>L - Lost</div>

<div>NR - No Result</div>

<div>NRR- Net Run Rate</div>

</td>

</tr>

</tfoot>

</table>

</body>

</html>

**16/04**

=====

Note: You can create nested table, but make sure that the child table is inside

<td> element.

Syntax:

<table>

<tr>

<td>

<table>

........

</table>

</td>

</tr>

</table>

Ex: Nested Tables

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<table width="100%" border="1" frame="void" cellspacing="10" cellpadding="10">

<tr height="400">

<td>

<table width="100%" border="0" frame="void" height="400">

<thead>

<tr>

<th colspan="2">Title</th>

</tr>

</thead>

<tbody>

<tr>

<td><img src="./images/a1.jpg" width="100%"></td>

<td> <img src="./images/a2.jpg" width="100%"></td>

</tr>

<tr>

<td><img src="./images/a3.jpg" width="100%"></td>

<td><img src="./images/a1.jpg" width="100%"></td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="2">More</td>

</tr>

</tfoot>

</table>

</td>

<td>

<table width="100%" border="0" frame="void" height="400">

<thead>

<tr>

<th colspan="2">Title</th>

</tr>

</thead>

<tbody>

<tr>

<td><img src="./images/a1.jpg" width="100%"></td>

<td> <img src="./images/a2.jpg" width="100%"></td>

</tr>

<tr>

<td><img src="./images/a3.jpg" width="100%"></td>

<td><img src="./images/a1.jpg" width="100%"></td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="2">More</td>

</tr>

</tfoot>

</table>

</td>

<td>

<table width="100%" border="0" frame="void" height="400">

<thead>

<tr>

<th colspan="2">Title</th>

</tr>

</thead>

<tbody>

<tr>

<td><img src="./images/a1.jpg" width="100%"></td>

<td> <img src="./images/a2.jpg" width="100%"></td>

</tr>

<tr>

<td><img src="./images/a3.jpg" width="100%"></td>

<td><img src="./images/a1.jpg" width="100%"></td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="2">More</td>

</tr>

</tfoot>

</table>

</td>

<td>

<table width="100%" border="0" frame="void" height="400">

<thead>

<tr>

<th colspan="2">Title</th>

</tr>

</thead>

<tbody>

<tr>

<td><img src="./images/a1.jpg" width="100%"></td>

<td> <img src="./images/a2.jpg" width="100%"></td>

</tr>

<tr>

<td><img src="./images/a3.jpg" width="100%"></td>

<td><img src="./images/a1.jpg" width="100%"></td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="2">More</td>

</tr>

</tfoot>

</table>

</td>

</tr>

</table>

</body>

</html>

www.telerik.com

**Forms in HTML**

- Form is a generic container.

- It provides an UI for user, so that user can view, input, edit or delete values in a data store.

- It allows to handle various interactions by using different elements like

button

textbox

checkbox

radio

listbox

dropdown etc.

- The <form> container is used to create an UI for handling **CRUD**.

C Create

R Read

U Update

D Delete

- Form is a generic container with validation, actions and methods pre-defined.

Syntax:

<form>

... elements ...

</form>

**Form Attributes:**

1. **id** ]

2. **name** ] used for reference in CSS, JavaScript etc.

3. **class** ]

Syntax:

<form id="frmLogin" name="LoginForm" class="form-group">

</form>

4. **method** : It defines the actions to be performed.

It defines the functionality to configure.

HTML form can be configured for 2 requests

a) GET

b) POST

**GET** submits to fetch data from server.

**POST** submits to save data on server.

GET vs POST

**GET**

- It is a request method to submit data for fetching.

- It submits data as "Query String".

- Query string is appended into URL and displayed in browser address bar.

page.html ?key1=value1&key2=value2

- **Query** is a key and value collection appended with ? mark and &.

- Everyone can view the values.

- It is easy to hack the values.

- It is not safe.

- It is stored in browser logs. [history]

- It can be bookmarked. [favorites]

- You can't submit complex data, like binary type data.

- There is a limit for data, max 2048 chars as per standards.

**POST**:

- It is a request method to submit and save data on server.

- It submits the data as "Form Body".

- It is not stored in browser logs.

- It can't be bookmarked.

- It is safe when compared to GET.

- It is not easy to hack your data.

- It allows complex formats of data.

- There is no limit for data.

**FAQ: If GET is not good for saving on server, then why form allows to save on GET?**

Ans: GET request can cache the data. Hence it can save round trips to server.

**Note: POST can't cache the data.**

**It is mandatory to define the target location while configuring POST action.**

5. **action** : It specifies the target location for submitting form data.

Typically it is a client side or server side page.

Syntax:

<form method="get" action="./page.html | php | jsp">

6. **novalidate** : It disables the default HTML 5 validations.

Syntax:

<form method="get" action="./page.html" novalidate>

**Form Elements**

**17/04**

=====

**Form Attributes**

- id

- name

- class

- method

- action

- novalidate

**Form Elements:**

**TextBox**

- It is an input that allows user to view, input and edit value.

- It is handles a string type data.

- String is a literal with group of characters like alphabet, number and special chars.

- Form <input> element allows to input a value.

- However HTML 5 introduces types for input data. So that it can set validation.

Syntax:

<input type="text"> => handle string

Attributes:

1. **id** ]

2. **name** ] used for reference

3. **class** ]

Note: A form can't submit the value of any element if "name" is not defined.

Every form element must have a name attribute.

<input type="text" name="txtName">

Names are defined using camel case. [ prefixSuffix ]

4. **value** : It specifies the default value to display in textbox.

Syntax:

<input type="text" name="txtName" value="john">

5. **readonly** : It will not allow to modify the value in textbox, but it allows to submit

the value.

6. **disabled** : It will not allow to modify & submit value.

Syntax:

<input type="text" name="txtName" value="John" disabled>

<input type="text" name="txtName" value="John" readonly>

7. **placeholder** : It defines the watermark text to display for textbox.

You can see placeholder only when value is not defined.

Syntax:

<input type="text" name="txtMobile" placeholder="+91 90000000011">

8. **autofocus** : It sets focus for textbox automatically on load.

Syntax:

<input type="text" name="txtMobile" autofocus>

9. **size** : It sets width for textbox, the default width is 20.

It is not limit for chars, it is just size of textbox.

Syntax:

<input type="text" name="txtOTP" size="5">

<input type="text" name="txtComments" size="40">

10. **maxlength** : It specifies the maximum limit of characters in textbox.

It sets restriction for number of characters.

Syntax:

<input type="text" name="txtOTP size="5" maxlength="5">

11. **minlength** : It sets validation for minimum number of chars in a textbox.

Syntax

<input type="text" name="txtName" minlength="4" maxlength="10">

- You can't type more than 10 chars : it is restriction

- You can type less than 4 chars button it will not accept : It is validation

12. **required** : It is used for mandatory fields. It will not allow to submit empty.

It is used to ensure that value is defined in a textbox.

Syntax:

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

Note: Minlength verifies the length of chars if entered. It will not verify the required

condition. Hence textbox needs both required and minlength.

13. **list** : It is used to configure a datalist for textbox.

Form datalist is a set of options that are displayed as auto complete text.

Syntax:

<input type="text" name="txtCourse" list="courses">

<datalist id="courses">

<option> Java </option>

<option> JavaScript </option>

</datalist>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Form</title>

</head>

<body>

<form>

<dl>

<dt>User Name</dt>

<dd><input type="text" name="txtName" required minlength="4" maxlength="10" placeholder="Name in block letters"></dd>

<dt>Mobile</dt>

<dd><input type="text" autofocus name="txtMobile" placeholder="US: +1(000) 0000-000"></dd>

<dt>OTP</dt>

<dd><input type="text" name="txtOTP" size="4" maxlength="4"></dd>

<dt>Your Course</dt>

<dd>

<input type="text" name="txtCourse" list="courses">

<datalist id="courses">

<option>Java</option>

<option>JavaScript</option>

<option>Oracle</option>

<option>AWS</option>

<option>UI Front End</option>

<option>UI Full Stack</option>

<option>Azure</option>

</datalist>

</dd>

</dl>

<button>Submit</button>

</form>

</body>

</html>

14. **pattern** : It is used to verify the format of input value.

Format is verified by using a regular expression.

Regular Expression is built by using meta characters & quantifiers.

**18/04**

=====

**Attributes for Textbox**

1. id

2. name

3. class

4. value

5. readonly

6.disabled

7. size

8. autofocus

9. minlength

10. maxlength

11. required

12. list

13. placeholder

14. pattern

**Meta character Description**

--------------------------------------------------------------------------------------------------------

**?** zero or one occurrence of a character.

Syntax:

pattern="colou?r" => color, colour

**\*** zero or more occurrences of a character.

Syntax:

pattern="colou\*r" => color, colour, colouur, colouuur..

**+** one or more occurrences of a character

Syntax:

pattern="colou+r" => colour, colouur, colouuur..

**. (dot)** any single character [alphabet, number or special]

Syntax:

pattern=".at" => cat, bat, 2at, $at ..

**| (or)** It refers to logical or in pattern where you can configure

multiple and it can match with any one of them.

Syntax:

pattern="red | green | blue" => red, green, blue

**\** It is used as an escape sequence character.

It can transform meta character to normal & vice versa.

Syntax:

pattern="gmail.com" => gmail2com, gmail$com...

pattern="gmail\.com" => gmail.com

**[ ]** It configures a set of values in random or range.

[a,d,s] => only specified chars allows [A,a,d,D,s,S] => both lower can uppercase of specified set

[a-m,A-M] => any char with in specified range

[0-9] => any numeric

[4-5] => any number with in specified range

[4,7,9] => only specified numbers

[a-z,A-Z,0-9] => all alphabet & numbers

[$,%,&,@] => only specified special chars allowed.

Syntax:

pattern= "[A-Z][0-9][a-z]" => B4u

pattern= "[0-9][0-9]" => 31, 24, 66

pattern="[0-9][0-9]?" => 3, 31

pattern="[0-9][A-Z]?[0-9] => 4A5, 52

**[^]** It is used to exclude the specified set of value.

It allows all other than the specified.

[^a,d,s] => a,d,s not allowed

[^A-Z, a-z, 0-9] => all special chars allowed

**( )** It is used for a union of expression.

**\d** It refers to any number from 0 to 9.

**\D** It excludes number and allow all others.

Syntax:

pattern="\D\d" => a5, A5, $5

pattern="\d\d?" => one or two digits number

**\w** It refers words chars, which include alphabet, number &

underscore. [a-z,A-Z,0-9, \_ ]

**\W** It refers to non-word chars. All special chars except "\_".

Syntax:

pattern="\W\d\D" $4A, $4a, $4#

**\s** It refers to blank space. You should not use a manual blank

space in pattern. It is always defined with "\s".

Syntax:

pattern="\d\s[A-Z]" 4 A, 5 H

**\i** It ignores the capitalization of text in expression.

Syntax:

pattern="colou?r\i" color, Color, Colour

**\^**  It refers expression starts with.

**$** It refers expression ends with.

Syntax:

pattern="\^ ....expression... $"

**Quantifier Description**

-----------------------------------------------------------------------------------------

{ 4 } exactly specified number of chars.

{ 4, 10 } minimum 4 and maximum 10 chars.

{ 4, } minimum 4 and maximum any chars.

Syntax:

pattern = "\d{10}" exactly 10 digits number

pattern = "\d{2}[A-Z]{3}\s\W{2}" 32HYD $#

Query: Write a pattern for Indian mobile format +91 and 10 digits number.

pattern= "\+91\d{10}"

pattern="\+91[0-9]{10}"

Query: Write a patter for US mobile format as shown below

+(1)(000) 000-0000

pattern="\+\(1\)\(\d{3}\)\s\d{3}-\d{4}"

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<form>

Pattern : <input type="text" placeholder="+(1)(000) 000-0000" pattern="\+\(1\)\(\d{3}\)\s\d{3}-\d{4}" name="pattern"> <button>Submit</button>

</form>

</body>

</html>

**19/04**

**Built-in patterns:**

**(?=.\*[A-Z])** At least one uppercase letter

**(?=.\*[0-9])** At least one numeric

**(?=.\*[a-z])** At least one lower case letter

**(?=.\*[!@#$%&])**  At least one special character

- You have to use the expression individually even when you are referring to multiple.

**(?=.\*[A-Z,0-9])** // Not Good

**(?=.\*[A-Z])(?=.\*[0-9])** // Good

- Avoid using meta characters like \d, \w etc.

(?=.\*[\d]) // Not Good

(?=.\*[0-9]) // Good

Query: Write a pattern to validate user name 4 to 15 chars alpha numeric.

Special chars not allowed. Uppercase & Lower case allowed.

Syntax:

pattern="[a-z,A-Z,0-9]{4,15}"

Query: Write a pattern to validate User name alpha numeric with \_ allowed.

between 4 to 14 chars. But it must start with uppercase letter.

Syntax:

pattern="[A-Z]\w{4,14}"

Query: Write a pattern to validate user name alpha number with underscore

4 to 15 chars with at least one uppercase letter.

Note: Always write the priority first.

Syntax:

pattern="(?=.\*[A-Z])\w{4,15}"

Query: Write a pattern to validate user name alpha numeric with underscore and

special chars. But at least one uppercase, one number & one special char

is required. [ 4 to 15 ]

Syntax:

pattern="(?=.\*[A-Z])(?=.\*[0-9])(?=.\*[\W])[\w,\W]{4,15}"

**Password Input**

- It is a string input that can mask the chars with "\*".

- All attributes are same as textbox.

- Only "list" attribute is not recommended for password.

Syntax:

<input type="password" name="Password" required pattern="">

**Number Input**

- It allows to input numeric value.

- It can set step value and range for number.

**Attributes:**

a) min

b) max

c) step

Syntax:

<input type="number" name="rate" min="10.45" max="20.45" step="0.01" value="12.45" required autofocus>

**Range Input**

- It similar to a number input, but contains a slider to select value.

- It will not allow to enter a value.

- It allows only to choose a value.

Syntax:

<input type="range" name="rate" min="1" max="100" value="50" step="5">

**Color Input**

- It shows a color picker that allows to select color from panel.

- Color value must be a 6 chars hexadecimal.

Syntax:

<input type="color" name="color" value="#ff0000">

**Email Input**

- It a string with default email validation.

- It ensures that the string contains "@" character.

Syntax:

<input type="email" autofocus required name="Email">

**URL Input**

- It validates a string in URL format.

- URL must have a protocol and proper domain extension.

www.amazon.in // invalid

https://www.amazon.in // valid

http://www.amazan.in // valid

Syntax:

<input type="url" name="URL" required autofocus>

**21/04**

- Textbox

- Password

- Number

- Range

- Email

- Url

- Color

**Date & Time Input**

- HTML provides following input types for handling date & time values

a) date

b) time

c) datetime-local

d) week

e) month

Syntax:

<input type="date">

<input type="datetime-local">

<input type="week">

- HTML date picker will not allow to select a range of dates.

- You can restrict the date selection by using "min & max" attributes.

- The date format for developer is always "year-month-day".

Syntax:

<input type="date" name="dept" min="2025-04-21" max="2025-05-09">

**File Input**

- It is a file browser.

- It allows to browse and select files on client device.

Syntax:

<input type="file" name="photo">

- You can set filter for specific file type by using "accept" attribute.

- Filter is not a permanent restriction for files, it can be changed dynamically.

Syntax:

<input type="file" name="photo" accept=".jpg">

<input type="file" name="resume" accept=".pdf, ".docx">

Note: You can restrict the file types, you can set only filters for file types.

- You can allow selection of multiple files by using "multiple" attribute.

Syntax:

<input type="file" name="photo" accept=".jpg" multiple>

**Radio Buttons**

- Radio allows to select one or multiple options from a group of choices.

- Radio once checked can't be unchecked.

- Developers use radios with "Mutex" mechanism. [ Mutual Exclusion ]

- Mutex configures multiple tasks to one thread.

- Hence process will allow only one task from a thread.

Syntax:

<input type="radio"> Text1 => both radios are individual

<input type="radio"> Text2

- To implement mutex you have to define same name for all radios in a category.

Syntax:

<input type="radio" name="gender"> Male

<input type="radio" name="gender"> Female

- You can make any radio selected by using "checked".

Syntax:

<input type="radio" name="gender" checked> Male

- Radios submit "ON" as default value.

- Hence every radio must be defined with a value attribute.

Syntax:

<input type="radio" name="gender" value="Male" checked> Male

- Value is an attribute used to submit content. It is not for UI caption.

- Radios and other form elements can have a bound caption by using "<Label>" element.

Syntax:

<input type="radio" name="gender" value="male"> <label for="gender"> Male </label>

- "for" is an attribute to bind label with relative form element. It must always map to name of element.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<form>

<dl>

<dt>Gender</dt>

<dd>

<input type="radio" checked value="Male" name="gender"> <label for="gender">Male</label>

<input type="radio" value="Female" name="gender"> <label for="gender">Female</label>

</dd>

</dl>

<button>Submit</button>

</form>

</body>

</html>

**Checkbox**

- It is similar to radio but allows to check and uncheck any option dynamically.

- It requires name & value.

- It can be selected using checked.

Syntax:

<input type="checkbox" name="course" value="HTML" checked>

<label for="course"> HTML </label>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.user {

border-left: none;

border-right: none;

border-top: none;

outline: none;

}

</style>

</head>

<body>

<form>

<dl>

<dt>User Name</dt>

<dd>

<input type="text" class="user" placeholder="User Name">

</dd>

<dt>Courses</dt>

<dd>

<input type="checkbox" checked value="HTML" name="Course"> <label for="Course"> HTML </label>

<br><br>

<input type="checkbox" checked value="CSS" name="Course"> <label for="Course"> CSS </label>

</dd>

</dl>

<button>Submit</button>

</form>

</body>

</html>

**Dropdown List**

- A dropdown is technically combo box.

- It provides a set of options and allows user to select any one from group of choices.

- The elements used for dropdown are

<select> configures a dropdown

<option> defines an item in dropdown

<optgroup> it groups a set of options under one label.

Syntax:

<select>

<option> Text </option>

<optgroup label="category-name">

<option> Item-1 </option>

<option> Item-2 </option>

</optgroup>

</select>

- Every option comprises of 3 attributes

a) value It defines the value to submit

b) selected It keeps the option selected on page load

c) disabled It will not allow to select option

Syntax:

<option value="to\_submit"> text\_to\_show </option>

<option selected> Text </option>

<option disabled> Text </option>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.user {

border-left: none;

border-right: none;

border-top: none;

outline: none;

}

</style>

</head>

<body>

<form>

<dl>

<dt>User Name</dt>

<dd>

<input type="text" class="user" placeholder="User Name">

</dd>

<dt>Courses</dt>

<dd>

<input type="checkbox" checked value="HTML" name="Course"> <label for="Course"> HTML </label>

<br><br>

<input type="checkbox" checked value="CSS" name="Course"> <label for="Course"> CSS </label>

</dd>

<dt>Departure</dt>

<dd>

<select name="Airport">

<option>Select City</option>

<optgroup label="International">

<option disabled>London</option>

<option>Newyork</option>

</optgroup>

<optgroup label="Domestic">

<option value="RGI">Hyderabad</option>

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

</optgroup>

</select>

</dd>

</dl>

<button>Submit</button>

</form>

</body>

</html>

**22/04**

=====

Datetime

Color

Range

Textbox

Password

Url

Email

Number

Radio

Checkbox

Dropdown

**List Box**

- You can transform a dropdown into list box by using the attributes

a) **size**

b) **multiple**

- List box allows to select one or multiple options from a group of choices.

Syntax:

<select size="3" multiple>

- Options and Select are RC type elements.

- RC type allows alphabet, number & special chars.

Syntax:

<select size="4" multiple>

<optgroup label="title">

<option> Text </option>

</optgroup>

</select>

Ex: Checkbox List

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.user {

border-left: none;

border-right: none;

border-top: none;

outline: none;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

ul {

list-style: none;

border: 1px solid gray;

padding: 10px;

width: 100px;

height: 50px;

overflow: auto;

}

</style>

</head>

<body>

<form>

<ul>

<li> <input type="checkbox"> <label>HTML</label> </li>

<li> <input type="checkbox"> <label>CSS</label> </li>

<li> <input type="checkbox"> <label>Bootstrap</label> </li>

<li> <input type="checkbox"> <label>jQuery</label> </li>

<li> <input type="checkbox"> <label>JavaScript</label> </li>

</ul>

</form>

</body>

</html>

**Textarea**

- It allows user to input multiline text.

- You can control textarea using

a) rows

b) cols

c) readonly

d) disabled

- It is RC type element, will not allow rich formats of text.

Syntax:

<textarea rows="4" cols="40" readonly | disabled>

... your text optional ...

</textarea>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.user {

border-left: none;

border-right: none;

border-top: none;

outline: none;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

</style>

</head>

<body>

<form>

<dl>

<dt>Your Comments</dt>

<dd><textarea rows="4" cols="40"></textarea></dd>

<dt>Terms of Service</dt>

<dd>

<textarea disabled rows="4" cols="40"> Lorem ipsum dolor sit amet consectetur adipisicing elit. Delectus officiis doloribus corporis enim voluptatum. Nostrum ut voluptatibus velit soluta voluptatum amet adipisci neque alias vel consequatur voluptate esse, consectetur vitae? Lorem ipsum dolor sit amet consectetur adipisicing elit. Saepe, nam autem quidem nostrum itaque earum temporibus, facere aut ex veniam mollitia esse quos tenetur minima maiores debitis blanditiis expedita totam. </textarea>

</dd>

<dd><input type="checkbox"> <label>I Accept</label></dd>

</dl>

<button>Post Comments</button>

</form>

</body>

</html>

**Meter & Progress Bar**

- **Meter** is used to show grade meter.

- **Progress** **bar** is used to display status of any task performed in page.

- Meter comprises of attributes

a) min

b) max

c) value

d) low

e) high

- If low and high value are set to zero or not defined then meter show green color.

Syntax:

<meter min="1" max="5" value="5"> </meter> Green

<meter min="1" max="5" value="5" low="0" high="0"> </meter> Green

- Low value must be above min and below max.

- High value must be above value and below max.

(or)

- High value must be above low and below max.

Note: If range between low & high is zero => Red

If range between low & high is less => Red

If range between low & high is more => yellow

if low & high are zero => Green

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.user {

border-left: none;

border-right: none;

border-top: none;

outline: none;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

ul {

list-style: none;

}

meter {

width: 200px;

height: 20px;

}

</style>

</head>

<body>

<form>

<ul>

<li> <span>5 <span class="bi bi-star-fill"></span> </span> <meter min="1" max="5" value="5"></meter> </li>

<li> <span>4 <span class="bi bi-star-fill"></span> </span> <meter min="1" max="5" value="3"></meter> </li>

<li> <span>2 <span class="bi bi-star-fill"></span> </span> <meter min="1" max="100" value="100" low="40" high="80"></meter> </li>

<li> <span>1 <span class="bi bi-star-fill"></span> </span> <meter min="1" max="100" value="100" low="60" high="80"></meter> </li>

</ul>

</form>

</body>

</html>

- Progress is similar to meter but can't configure colors.

- It is a standard progress of actions performed.

Syntax:

<progress min="1" max="100" value="40"> </progress> 40% completed

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.user {

border-left: none;

border-right: none;

border-top: none;

outline: none;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

progress {

width: 200px;

height: 20px;

}

</style>

</head>

<body>

<form>

<dl>

<dt>Preparing for download</dt>

<dd><progress></progress></dd>

<dt>40% Downloaded</dt>

<dd>

<progress min="1" max="100" value="40"></progress>

</dd>

</dl>

</form>

</body>

</html>

**Buttons**

- Buttons are used to confirm user actions.

- Buttons in HTML are categorized into 2 types

a) Generic Buttons

b) Non Generic Buttons

- Generic Buttons have predefined functionality.

a) HTML 4

<input type="submit">

<input type="reset">

b) HTML 5

<button type="submit"> Text | Image | Icon </button>

<button type="reset"> Text | Image | Icon </button>

Note: Submit allows to submit the form data as query string on GET and

form body on POST.

Reset allows to reset a form to the default state.

- Non Generic button are static buttons without any pre-defined functionality

a) HTML 4

<input type="button" value="any">

b) HTML 5

<button type="button"> Text | Image | Icons </button>

Note: If button type is not defined in HTML 5, the default is submit type.

<button> Text </button> => Submit

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.user {

border-left: none;

border-right: none;

border-top: none;

outline: none;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

progress {

width: 200px;

height: 20px;

}

</style>

</head>

<body>

<form>

<dl>

<dt>Name</dt>

<dd><input type="text" name="Name"></dd>

<dd>

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

<input type="reset" value="Cancel">

<input type="button" value="Print">

</dd>

<dd>

<button type="submit"> <span class="bi bi-person-fill"></span> Register </button>

<button type="reset"> <img src="./images/a5.jpg" width="40" height="20"> Cancel</button>

<button type="button"> <span class="bi bi-printer-fill"> Print </span> </button>

</dd>

</dl>

</form>

</body>

</html>

Ex: Login Form

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.user {

border-left: none;

border-right: none;

border-top: none;

outline: none;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

body {

display: flex;

justify-content: center;

font-family: Arial;

}

form {

width: 300px;

border:1px solid gray;

padding: 30px;

border-radius: 5px;

}

.title {

font-weight: bold;

font-size: 30px;

}

.form-group {

margin-top: 15px;

margin-bottom: 15px;

}

.form-label {

font-weight: bold;

margin-bottom: 5px;

display: block;

}

.form-control, .form-select {

font-size: 20px;

width: 95%;

padding: 5px;

}

.btn-login {

background-color: gold;

border:none;

font-size: 20px;

padding: 10px;

width: 100%;

border-radius: 10px;

}

a {

text-decoration: none;

}

</style>

</head>

<body>

<form>

<div class="bi bi-person-fill title"> User Login </div>

<div class="form-group">

<label class="form-label" for="UserName"> User Name </label>

<div>

<input type="text" placeholder="Name in Block Letters" name="UserName" class="form-control">

</div>

</div>

<div class="form-group">

<label class="form-label" for="Password"> Password </label>

<div>

<input type="password" placeholder="One Uppercase letter" name="Password" class="form-control">

</div>

</div>

<div class="form-group">

<label class="form-label" for="City"> Your City </label>

<div>

<select class="form-select" name="City">

<option>Select Your City</option>

<option>Delhi</option>

<option>Hyd</option>

</select>

</div>

</div>

<div class="form-group">

By continuing, you agree to Amazon's <a href="#">Conditions of Use</a> and <a href="#">Privacy Notice</a>.

</div>

<div>

<button type="submit" class="btn-login">Login</button>

</div>

</form>

</body>

</html>

**CSS Form Classes**

**23/04**

=====

**CSS Form Classes**

**Element State Classes:**

:hover on mouse over

:active on mouse down

:focus on focus [ element gets focus when you click inside ]

:read-only when read-only attribute is set

:disabled when disable attribute is set

:checked when radio or checkbox is checked

:target when element is target of any hyperlink

:visited when link is in visited state

**FAQ: What is a class?**

Ans: Class is a program template in computer programming.

You can customize and implement according to requirements.

**Ex-1: hover, active**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

button {

border:none;

border-radius: 5px;

background-color: rgb(202, 202, 0);

}

button:hover {

background-color: yellow;

cursor: grab;

}

button:active {

box-shadow: 3px 3px 2px black;

}

</style>

</head>

<body>

<button> Login </button>

</body>

</html>

**Ex-2: Focus**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.UserName+div {

display: none;

}

.UserName:focus+div {

display: block;

font-size: 12px;

color:red;

}

.UserName:focus {

border:1px solid red;

box-shadow: 2px 2px 2px red;

outline: none;

margin-bottom: 5px;

}

</style>

</head>

<body>

<dl>

<dt>User Name</dt>

<dd>

<input type="text" value="John" class="UserName" placeholder="Name in Block Letters">

<div>Name in Block Letters</div>

</dd>

</dl>

</body>

</html>

**Ex-3: read-only, disabled**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

input:read-only {

cursor: not-allowed;

background-color: lightgray;

}

input:disabled {

background-color: lightyellow;

}

</style>

</head>

<body>

<dl>

<dt>User Name</dt>

<dd>

<input type="text" readonly value="John" class="UserName">

</dd>

<dt>Mobile</dt>

<dd>

<input type="text" disabled value="+9198765554321" class="Mobile">

</dd>

</dl>

</body>

</html>

**Ex-4: Checked**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.terms+label {

color:red;

}

.terms:checked+label {

color:green;

}

.terms~button {

display: none;

}

.terms:checked~button {

display: block;

}

.preview+img {

display: none;

}

.preview:checked+img {

display: block;

}

</style>

</head>

<body>

<dl>

<input type="checkbox" class="preview"> Preview

<img src="./images/iphone-white.jpg" width="100" height="100">

<dt>Terms of Service</dt>

<dd>

<textarea rows="4" cols="40" disabled>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Qui enim repellendus voluptatem aspernatur minima eveniet distinctio reiciendis, libero vitae mollitia et exercitationem tempora, dolores quibusdam ratione nulla odit! Nam, neque. Lorem ipsum dolor, sit amet consectetur adipisicing elit. At aspernatur repudiandae illum corporis atque quis, in nobis similique iusto officiis. Sapiente, vel magni. Quod repellat nesciunt voluptate quisquam harum facilis.</textarea>

</dd>

<dd>

<input type="checkbox" class="terms"> <label> I Accept </label>

<br><br>

<button>Submit</button>

</dd>

</dl>

</body>

</html>

**Validation State Classes:**

:required when element set with required attribute

:optional when element is not having required attribute

:in-range when value is with in specified range

:out-of-range when value is not within the specified range.

:valid on validation pass

:invalid on validation fail

Note: "required" class is just to verify weather the attribute is defined or not.

"required" satisfied on not is verified by using "invalid & valid" class.

Input elements can use "placeholder" class for configuring placeholder effects.

::placeholder

Syntax:

input :: placeholder {

// styles

}

**Multimedia**

**24/04**

=====

**Multimedia Elements**

- Multimedia refers to animation, audio and video content.

- HTML provides elements for handling multimedia

a) <marquee> [ obsolete - deprecated ]

b) <audio>

c) <video>

d) <embed> [ obsolete for video & audio ]

Marquee:

- It sets scrolling and sliding content in page.

- You can control speed, direction and style.

Syntax:

<marquee> ... your content ... </marquee>

**Attributes:**

1. scrollamount : It controls marquee scrolling speed from 1 to 100.

2. direction : You can set scrolling direction left, right, up & down.

3. behavior : It changes the scrolling to sliding using "alternate" as value.

4. width & height : It sets width and height for marquee area.

5. bgcolor : It sets background color for marquee area

6. loop : It specifies the number of times to display.

Syntax:

<marquee scrollamount="20" loop="3" direction="right" width="400" bgcolor="yellow">

.... your content ....

</marquee>

<marquee behavior="alternate"> </marquee>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<marquee scrollamount="20" loop="3" bgcolor="yellow">

Summer Sale Stars 1<sup>st</sup> May 2025 <img width="50" height="50" src="./images/iphone-white.jpg">

</marquee>

</body>

</html>

**Video & Audio:**

- HTML supports video of type

a) avi

b) mp4

c) mkv

d) mov

e) ogg etc..

- HTML support audio of type

a) mp3

b) wav

c) midi etc..

- It is not recommended to embed them directly into page in modern web designing.

- You can upload into any podcasting site and embed its content into your page.

Syntax:

<audio src="" width="" height="" controls poster="" autoplay> </audio>

<video src="" .... > </video>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<video src="./videos/js-class.mp4" poster="./images/data sci.png" controls width="300" height="200"></video>

</body>

</html>

**25/04**

=====

1. **Inline**

2. **Embedded**

- Where to embed

- Can we have multiple style containers

- If multiple are defined then how priority works.

- MIME type

- Media

3. **External File**

- You can configure styles in a separate style sheet.

- It can cascade over HTML.

- Hence they are known as Cascading Style Sheets.

- You can implement styles across pages.

- Using an external style sheet will increase the number of requests for a page, which increases the page load time.

Ex:

1. Add a new style sheet into "src/styles/home.css"

2. style attributes are directly defined as style object.

h1

{

color: red;

}

3. Link the style sheet to your HTML page.

<link rel="stylesheet" href="../src/styles/home.css">

Ex:

home.css

.ticket {

background-color: yellow;

padding: 20px;

border:4px double red;

}

dt {

font-weight: bold;

background-color: red;

color:white;

width: 200px;

}

home.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../src/styles/home.css">

</head>

<body>

<div class="ticket">

<h2>Your Ticket</h2>

<aside>

<img src="./images/a1.jpg" width="100" height="100">

</aside>

<dl>

<dt>Departure</dt>

<dd>Hyderabad</dd>

<dt>Arrival</dt>

<dd>Delhi</dd>

<dt>Date</dt>

<dd>25-April-2025</dd>

</dl>

<button onclick="window.print()">Print Ticket</button>

</div>

</body>

</html>

FAQ: If styles are defined using all three techniques the which set of styles will apply?

Ans:

1st Priority Inline

2nd Priority Depends on which one is last configured.

If same attribute is defined, then it uses priority.

If different attributes are defined then all will apply.

**FAQ: What is minification?**

Ans: Minification is a coding technique followed by developers to reduce the file size.

It is the process of removing optional code snippets, line breaks, blank spaces

and using all possible shortcuts to implement functionality.

https://www.toptal.com/developers/cssminifier

**FAQ: What is difference between uncompressed and minified?**

Ans : Uncompressed is heavy and occupies more memory.

Minified is light weight and faster in render.

Uncompressed is good for Development environment.

Minified is good for Production environment. [ Go Live ]

Ex:

1. Copy your actual CSS code

2. Go to the minifier website " https://www.toptal.com/developers/cssminifier "

3. Paste the actual code

4. Click Minify button

5. Copy the minified code

6. Go to your project and add a new file "home.min.css"

7. Paste minified code

8. Link the minified file to your page.

<link rel="stylesheet" href="../src/styles/home.min.css">

**CSS Selectors**

- Selectors are required when you are configuring styles embedded or in external file.

- They are used to select your HTML elements to apply specified styles.

Syntax:

<h1 style="color:red"> without a selector

<style>

selector

{

color: red;

}

</style>

- CSS provides various types of selectors to select HTML elements.

- Officially CSS doesn't provide any set of categories for selectors however the popularly known categories in developers community are

1. Primary Selectors

2. Relational or Rational Selectors

3. Attribute Selectors

4. Dynamic Pseudo Classes

5. Structural Classes

6. Element State Classes

7. Validation State classes

8. Universal Selector

9. Root Selector

10. Language Selector etc..

**Primary Selectors:**

1. Type Selector

2. ID Selector

3. Class Selector

**Type Selector:**

- It refers to the element name.

- It configures styles to every occurrence of element in page.

- You can't ignore styles for any specific occurrence.

Syntax:

h1, p, div, span {

}

**ID Selector:**

- Every element can have a reference ID.

- CSS can access the ID and configure styles.

- You can choose elements that have to use the specified styles.

Syntax:

#text-style

{

color: red;

}

<h1 id="text-style">

<p id="text-style">

<div id="text-style">

- Every element can have only one ID reference.

- You can't define multiple categories styles to one element using ID selector.

**Class Selector:**

- Class is configure with a set of styles under reference name starting with "."

- Every element can implement multiple classes.

- You can apply multiple categories of styles for one element.

Syntax:

.text-style {

}

<h2 class="text-style">

- Multiple classes a separated with space delimiter.

<h2 class="class1 class2 class3 ..">

**28/04**

=====

**Primary Selectors**

- Type Selector

- ID

- Class

**FAQ: If styles are defined using all three primary selectors, then which set will apply?**

Ans:

1st Priority ID selector

2nd Class selector

3rd Type selector

If different styles attributes are defined then all will apply.

**FAQ: If ID selector is defined embedded and form external file, then which one will**

apply?

Ans: Which ever style is latest that will apply. LIFO [Last In-First Out]

**2. Rational / Relational Selectors**

Child selector parent child { }

Direct Child Selector parent > child { }

Adjacent Selector parent + adjacent { }

All Siblings parent ~ sibling { }

- Child selector applies styles to all child elements at level of hierarchy inside parent.

.container p { }

- Direct child applies styles to only direct elements not any nested element.

.container > p { }

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.container > p {

color:red;

}

</style>

</head>

<body>

<div class="container">

<p>Para-1</p>

<p>Para-2</p>

<div><p>Para-3</p></div>

<p>Para-4</p>

</div>

</body>

</html>

- Adjacent select immediately after the specified

parent + adj { }

- Siblings select all occurrences after the specified but within specified scope.

parent ~ sibling { }

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

h1~p {

color:red;

}

</style>

</head>

<body>

<div>

<h1>Heading</h1>

<p>Para-1</p>

<p>Para-2</p>

<p>Para-3</p>

<p>Para-4</p>

<p>Para-5</p>

</div>

<p>Para-6</p>

</body>

</html>

**3. Attribute Selectors**

- It allows to select element based on attributes and their value.

- You can select by only attribute or by using both attribute & value.

Syntax: Verifying only attribute

token[attribute] only attribute

Syntax: Verifying both attribute & value

**token[attribute=value]**

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

input[type="button"] {

background-color: yellow;

}

</style>

</head>

<body>

<form>

<dl>

<dt>User Name</dt>

<dd><input type="text"></dd>

<dt>Mobile</dt>

<dd><input type="text"></dd>

</dl>

<input type="button" value="Submit">

<input type="button" value="Cancel">

</form>

</body>

</html>

EX:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

p[id] {

color:red;

}

</style>

</head>

<body>

<p id>Para-1</p>

<p>Para-2</p>

<p id="p3">Para-3</p>

<p>Para-4</p>

<p id="p5">Para-5</p>

<p>Para-6</p>

</body>

</html>

**Attribute Conditions:**

[attribute=value] exact match

[attribute^=value] starts with

[attribute$=value] ends with

[attribute|=value] must start with specified and can have "-"

delimiter when defined with others.

[attribute~=value] any occurrence separated with blank space.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

p[class~="Effects"] {

color:red;

}

</style>

</head>

<body>

<p class="Effects">Para-1</p>

<p class="ParaEffects">Para-2</p>

<p class="Para-Effects">Para-3</p>

<p class="Effects-Para">Para-4</p>

<p class="Para Effects Text">Para-5</p>

<p class="Text Para Effects">Para-6</p>

<p class="Effects Para">Para-7</p>

<p class="Para Effects">Para-8</p>

</body>

</html>

Note: You have group the selectors.

CSS will not allow to group attributes.

Syntax:

select[size], select[multiple]

{

}

select[size, multiple] // invalid

{

}

p, h1, div { // selects all to define same style

}

p, .bg-style, #text-style // select all specified

{

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

select[size], select[multiple] {

font-size: 20px;

background-color: yellow;

}

</style>

</head>

<body>

<select>

<option>Delhi</option>

<option>Hyd</option>

</select>

<select size="2">

<option>Delhi</option>

<option>Hyd</option>

</select>

<select multiple>

<option>Delhi</option>

<option>Hyd</option>

</select>

</body>

</html>

**4. Structural Pseudo Classes**

:first-child first child element

:last-child last child element

:nth-child() specific child or even & odd occurrences

:nth-of-type() It can configure every nth-occurrence from start

:nth-last-of-type() It can configure every nth from end.

Syntax:

li : nth-child(3) { } 3rd element

li : nth-child(even){ } all even occurrences

li : nth-of-type(3n){ } every third start from top 3rd.

li : nth-of-type(3n+2){ } every third starting from top 2nd.

li : nth-last-of-type(3n+2){} every third starting from bottom 2nd.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

ul {

list-style: none;

}

li {

width: 100px;

border:1px solid gray;

padding: 5px;

margin-top: 10px;

margin-bottom: 10px;

}

li:nth-last-of-type(3n+2) {

background-color: yellow;

}

</style>

</head>

<body>

<ul>

<li>Item-1</li>

<li>Item-2</li>

<li>Item-3</li>

<li>Item-4</li>

<li>Item-5</li>

<li>Item-6</li>

<li>Item-7</li>

<li>Item-8</li>

<li>Item-9</li>

<li>Item-10</li>

</ul>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

tbody tr:nth-child(even){

background-color: rgb(144, 224, 144);

}

tbody tr:nth-child(odd){

background-color: rgb(217, 252, 217);

}

thead, tfoot tr {

background-color: green;

color:white;

}

</style>

</head>

<body>

<table border="1" width="100%">

<thead>

<tr>

<th colspan="2">Name</th>

<th colspan="3">Address</th>

</tr>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>City</th>

<th>State</th>

<th>Postal Code</th>

</tr>

</thead>

<tbody>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td rowspan="7" align="center"> D<br>E<br>L<br>H<br>I </td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="5" align="center">&copy; copyright 2025</td>

</tr>

</tfoot>

</table>

</body>

</html>

**5. Dynamic Element State Classes**

**6. Validation State Classes**

**7. Behavioral and occurrence classes**

::first-letter

::first-line

::before

::after

::placeholder

::selection

CSS Units, CSS Colors, Inheritance, Rules

**29/04**

=====

**8. Universal Selector**

- It is used to select all elements in page.

\* {

}

- "\*" is a meta character that refers to zero or more occurrences of element.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

\* {

font-family: Arial;

}

</style>

</head>

<body>

<h1>Heading</h1>

<p>Para</p>

<div>Div</div>

<span>Span</span>

<input type="checkbox"> <label>Checkbox</label>

</body>

</html>

**9. Root Selector**

- It is used to select the root element in shadow DOM.

Syntax:

:root {

// styles

}

**10. Language Selector**

- It is used to configure styles based on language.

- It is applicable to multilingual pages.

Syntax:

:lang("name")

{

// styles

}

Summary

- CSS Integration

- CSS Selectors

- CSS Rules

**CSS Inheritance**

- Inheritance is a mechanism where the child elements implements some of the parent features.

- You can control the inheritance by using following CSS values

a) inherit

b) initial

c) unset

- "inherit" implements the parent styles.

- "initial" ignores the parent styles.

- "unset" ignores or removes the default styles.

Syntax:

child {

attribute : inherit;

attribute: initial;

attribute: unset;

}

child {

all : inherit | initial | unset;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.container {

border: 3px double red;

padding: 20px;

color:red;

}

h1 {

border:inherit;

padding: inherit;

color:initial;

font-weight: unset;

}

</style>

</head>

<body>

<div class="container">

Container Text

<h1>Welcome</h1>

</div>

</body>

</html>

**CSS Colors**

- CSS can configure colors using

1. color name

2. shade name

3. hexadecimal

4. rgb()

5. rgba()

6. hsl()

7. hsla()

8. linear-gradient ]

9. radial-gradient ] background-image

**CSS Units**

- Units are requires for measurement.

- You can control the size, orientation and location using units.

- CSS units are categorized into 2 types

a) Absolute Units

b) Relative Units

- Absolute units doesn't depend on other elements in page.

- CSS absolute units are

cm

mm

Q

in

pc

pt

px

- Relative units depend on relative elements like parent, root or adjacent etc.

- CSS relative units are

a) em [element]

b) rem [root element]

c) %

d) vh [viewport height]

e) vw [viewport width]

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.container {

font-size: 40px;

border: 1px solid black;

padding: 10px;

}

.msg {

font-size: 0.5em;

}

.body-msg {

font-size: 0.5rem;

}

</style>

</head>

<body>

body text

<div class="container">

Container text

<div class="msg">Div in container</div>

<div class="body-msg">Body relative text</div>

</div>

</body>

</html>

Note: "em & rem" are used relative to chars size, and not the container size.

You have to use "%" that sets by ratio of parent.

Syntax:

.child {

width : 50%;

height: 50%;

}

Note: "vh & vw" depends on the current view port height and width.

Viewport refers to device screen height and width.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.parent {

width: 200px;

height: 200px;

border: 3px solid red;

padding: 10px;

}

.child {

border:2px dotted blue;

height: 50vh;

width: 50vw;

}

</style>

</head>

<body>

<div class="parent">

<div class="child">

</div>

</div>

</body>

</html>

Summary

- CSS Integration

- CSS Selectors

- CSS Units

- CSS Colors

- CSS Rules

- CSS Inheritance

**CSS Box Model**

1. Margins

2. Padding

3. Border

4. Border Radius

5. Width

6. Height

**Margin Shorthand**

margin: top right bottom left ;

{

margin: 20px 0px 20px 0px;

}

{

margin : 20px;

}

**Padding Shorthand**

padding: top right bottom left;

{

padding : 20px 0px 20px 0px;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.parent {

border: 4px solid black;

margin : 100px 50px 0px 50px;

padding: 40px 0px 40px 0px;

}

</style>

</head>

<body>

<div class="parent">

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quidem libero deserunt deleniti nihil dolorem nesciunt quam quaerat quisquam exercitationem itaque, consequuntur nam enim expedita quasi voluptatibus harum fugiat tempora suscipit?

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Explicabo similique, praesentium unde cupiditate aliquam blanditiis alias, earum accusantium itaque veniam iusto architecto officiis, reprehenderit iure labore sit at dolore soluta.

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quisquam, alias error aut laudantium perferendis quia delectus obcaecati consequuntur sit, dolor omnis reprehenderit optio voluptatum! Quas temporibus reprehenderit minima magni quasi.

</div>

</body>

</html>

30/05

=====

CSS Box Model

1. Margins

2. Padding

3. Border

4. Border Radius

5. Width

6. Height

7. Border Image

**Border Styles:**

border : short hand for style, size and color

border-style : double, dotted, solid, groove, dashed ..

border-width : size in pixels or other units

border-color : any CSS color

border-left : short hand for left

border-left-style

border-left-width

border-left-color

... similarly for other directions...

**Border Radius:**

border-radius : all directions

border-top-left-radius

border-top-right-radius

border-bottom-left-radius

border-bottom-right-radius

**Border Image:**

border-image : It sets URL, offset and style.

Style can be stretch or space.

Offset can be a value between 1 to 100.

Note: You can apply border image, only when border is set.

Syntax:

{

border : 10px solid transparent;

border-image: url("path") stretch | space 70;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.parent {

border : 20px solid transparent;

border-image: url("../public/images/border.gif") stretch 80;

padding: 30px;

}

</style>

</head>

<body>

<div class="parent">

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quidem libero deserunt deleniti nihil dolorem nesciunt quam quaerat quisquam exercitationem itaque, consequuntur nam enim expedita quasi voluptatibus harum fugiat tempora suscipit?

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Explicabo similique, praesentium unde cupiditate aliquam blanditiis alias, earum accusantium itaque veniam iusto architecto officiis, reprehenderit iure labore sit at dolore soluta.

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quisquam, alias error aut laudantium perferendis quia delectus obcaecati consequuntur sit, dolor omnis reprehenderit optio voluptatum! Quas temporibus reprehenderit minima magni quasi.

</div>

</body>

</html>

**CSS Positions**

1. Static

2. Absolute

3. Relative

4. Fixed

5. Sticky

Static

- It keeps element according to normal flow of document.

- It will not allow element to move using top, left, right & bottom attributes.

- It is the default position for element.

**Absolute:**

- It removes element from normal flow of document.

- It allows to move element using left, right, top and bottom.

- It keeps element with regard to content in page.

- It moves along with the content.

**Fixed:**

- It removes element from normal flow of document.

- It allows to move element using left, right, top and bottom.

- It keeps element with regard to browser window not with content in page.

**Sticky:**

- It keeps element according to normal flow of document.

- It allows to scroll up to specified units, and can lock scrolling after reaching the designated position.

**Relative:**

- It is defined for parent element.

- It allows to keep child element with regard to the parent.

- It requires child elements configured with "absolute" position only.

Syntax:

.parent {

position : relative;

}

.child {

position : absolute;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.men-fashion {

background-image: url("../public/images/men-fashion.jpg");

background-size: cover;

width: 600px;

height: 400px;

border:1px solid blue;

position: relative;

}

.offer{

background-color: red;

color:white;

width: 80px;

height: 80px;

border-radius: 80px;

text-align: center;

padding: 10px;

font-size: 28px;

font-weight: bold;

position: absolute;

top: 20px;

right: 20px;

}

nav {

margin: 30px 0px 30px 0px;

}

button {

font-size: 25px;

padding: 10px;

position: relative;

border:none;

background-color: yellow;

border-radius: 10px;

}

.badge {

display: inline-block;

width: 15px;

height: 15px;

background-color: red;

color:white;

padding: 5px;

font-size: 12px;

border-radius: 20px;

position: absolute;

top: -10px;

right: -10px;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body>

<nav>

<button class="bi bi-cart4"> <span class="badge">2</span> </button>

</nav>

<div class="men-fashion">

<div class="offer">Save<br>40%</div>

</div>

</body>

</html>

**CSS Z-Index**

- It is used to control the overlapping elements.

- It can send back or backward.

- It can bring front or forward.

- Index starts with "0" which refers to element at bottom.

Syntax:

{

z-index: 0;

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.women-fashion {

background-image: url("../public/images/women-fashion.jpg");

background-size: cover;

width: 500px;

height: 300px;

z-index: 0;

}

.men-fashion {

background-image: url("../public/images/men-fashion.jpg");

background-size: cover;

width: 400px;

height: 200px;

top: 150px;

left: 400px;

z-index: 2;

}

.kids-fashion {

background-image: url("../public/images/kids-fashion.jpg");

background-size: cover;

width: 300px;

height: 200px;

left: 300px;

top: 0px;

z-index: 1;

}

div {

box-shadow: 5px 5px 2px black;

border-radius: 20px;

position: absolute;

transition: 5s;

}

div:hover {

z-index: 2;

left: 600px;

transition: 5s;

}

</style>

</head>

<body>

<div class="women-fashion">

</div>

<div class="men-fashion">

</div>

<div class="kids-fashion">

</div>

</body>

</html>

**CSS Display**

1. **none** : It hides the element and removes allocated space from page.

**FAQ: To hide element in page we can use "visibility: hidden", then what is difference between display & visibility ?**

Ans: Visibility will hide element without removing the allocated space.

Display will hide element and remove allocated space.

2. **block** : It transforms an inline element into block level element.

3. **inline** : It transforms an block level element into inline element.

4. **inline-block** : It keeps element in same line and configures block style to

defines various styles.

Note: Few elements can't adapt the styles if they are not block level.

Hence you have to define "inline-block".

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

span {

display:inline-block;

background-color: yellow;

width: 100px;

text-align: center;

}

div {

display: inline;

}

</style>

</head>

<body>

<div>Welcome</div> to CSS.

<span>Home</span> Welcome to Home Page.

</body>

</html>

5. **flex**

**1/05**

====

**CSS Display**

- none

- inline

- block

- inline-block

5. **flex** : It is used for a container to keep all its elements flexible by row or

column. Flex have various supporting attributes

a) flex-direction

b) flex-wrap

c) justify-content

d) align-items

- **Flex direction** can be row, row-reverse, column, & column-reverse

- **Justify Content** can be space-between, around, evenly & center.

- **Align Items** depends on direction

a) row : baseline [top], center, end [bottom]

b) column: baseline [left], center, end [right]

- **Flex wrap** allows to wrap elements into multiple rows and columns.

You can add wrapping or remove to display compressed.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

nav {

border: 1px solid black;

padding: 20px;

height: 400px;

display: flex;

flex-direction: column;

flex-wrap: wrap;

align-items: center;

justify-content:space-between;

}

nav div {

width: 100px;

border:1px solid gray;

padding: 5px;

text-align: center;

}

</style>

</head>

<body>

<nav>

<div>Home</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>Home</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>Home</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>Home</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>Home</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>Home</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>Home</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

<div>About</div>

<div>Contact</div>

<div>Shop</div>

<div>Blog</div>

</nav>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

ul {

list-style: none;

display: flex;

flex-direction: row;

}

li {

border:1px solid black;

padding: 10px;

width: 100px;

display: flex;

flex-direction: row;

justify-content: space-around;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body>

<ul>

<li><span class="bi bi-house"></span> Home</li>

<li><span class="bi bi-globe"></span> About</li>

<li><span class="bi bi-envelope"></span> Contact</li>

<li><span class="bi bi-webcam"></span> Blog</li>

</ul>

</body>

</html>

**6. Grid**

- It is used for creating a responsive layout.

- Layout comprises data arranged in rows and columns.

- Grid can dynamically control the data in rows and columns.

- Grid attributes are

a) grid-template-columns

b) grid-row

c) grid-column

d) grid-gap

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

header, nav, main, aside, footer {

border:1px solid black;

padding: 10px;

margin: 5px;

}

main {

height: 400px;

}

body {

display: grid;

grid-template-columns: 2fr 8fr 2fr;

}

header {

grid-row: 1;

grid-column: 1/4;

background-color: yellow;

}

nav {

grid-row: 2;

grid-column: 1;

background-color: lightgreen;

}

main {

grid-row: 2;

grid-column: 2;

}

aside {

grid-row: 2;

grid-column: 3;

background-color: lightcyan;

}

footer {

grid-row: 3;

grid-column: 1/4;

}

</style>

</head>

<body>

<header>

Header

</header>

<nav>Nav</nav>

<main>Main</main>

<aside>Aside</aside>

<footer>Footer</footer>

</body>

</html>

Note: You have to use "**media queries**" to know the screen or device size, so that

content can be adjusted according to device.

Syntax:

@media type (condition)

{

}

type : It refers to screen, print

condition : It uses min-width, max-width, orientation.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

header, nav, main, aside, footer {

border:1px solid black;

padding: 10px;

margin: 5px;

}

main {

height: 400px;

}

@media screen and (orientation:landscape){

body {

display: grid;

grid-template-columns: 2fr 8fr 2fr;

}

header {

grid-row: 1;

grid-column: 1/4;

background-color: yellow;

}

nav {

grid-row: 2;

grid-column: 1;

background-color: lightgreen;

}

main {

grid-row: 2;

grid-column: 2;

}

aside {

grid-row: 2;

grid-column: 3;

background-color: lightcyan;

}

footer {

grid-row: 3;

grid-column: 1/4;

}

}

@media screen and (orientation:portrait) {

body {

display: grid;

grid-template-columns: 12fr;

}

header {

grid-row: 1;

background-color: yellow;

}

nav {

grid-row: 2;

background-color: lightgreen;

}

main {

grid-row: 3;

}

aside {

grid-row: 4;

background-color: lightcyan;

}

footer{

grid-row: 5;

}

}

</style>

</head>

<body>

<header>

Header

</header>

<nav>Nav</nav>

<main>Main</main>

<aside>Aside</aside>

<footer>Footer</footer>

</body>

</html>

Ex: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

header {

font-size: 20px;

font-family: Arial;

padding: 20px;

}

.brand-title {

font-size: 24px;

font-weight: bold;

}

@media screen and (orientation:landscape){

header {

display: flex;

flex-direction: row;

justify-content: space-between;

}

nav span {

margin-right: 30px;

}

.btn {

display: none;

}

.women-fashion {

background-image: url("../public/images/women-fashion.jpg");

background-size: cover;

width: 300px;

height: 300px;

}

}

@media screen and (orientation:portrait) {

nav {

display: none;

}

.btn {

display: inline;

font-size: 24px;

}

header {

display: flex;

justify-content: space-between;

}

.women-fashion {

background-image: url("../public/images/girl-window.jpg");

background-size: cover;

width: 100%;

height: 300px;

}

}

</style>

</head>

<body>

<header>

<div>

<span class="brand-title">Shopper.</span>

</div>

<nav>

<span>Home</span>

<span>Shop</span>

<span>Pages</span>

<span>Blog</span>

<span>Docs</span>

</nav>

<div class="btn">

<span class="bi bi-justify"></span>

</div>

</header>

<main>

<div class="women-fashion">

</div>

</main>

</body>

</html>

2/05

===

**Ex: Responsive Shopper Template**

1. src/styles/shopper-template.css

.header-row-1 {

background-color: #e6e4e4;

padding: 15px;

font-family: Arial;

font-size: 14px;

display: flex;

flex-direction: row;

justify-content: space-between;

}

.header-row-1 span {

padding-left: 10px;

padding-right: 10px;

}

.header-row-2 {

display: flex;

flex-direction: row;

justify-content: space-between;

align-items: center;

padding: 10px;

font-size: 20px;

font-family: Arial;

}

.brand-title {

font-size: 30px;

font-weight: bold;

}

nav span {

padding-left: 15px;

padding-right: 15px;

font-size: 18px;

}

.short-cuts span {

padding-left: 5px;

padding-right: 5px;

}

.bi-justify {

display: none;

}

article {

background-color: black;

color:white;

padding: 15px;

text-align: center;

font-family: Arial;

font-size: 16px;

}

.bi-lightning-fill {

color:gold;

}

main {

height: 450px;

display: grid;

grid-template-columns: 4fr 4fr 4fr;

margin-top: 10px;

}

.women-fashion {

background-image: url("../../public/images/women-fashion.jpg");

background-size: cover;

display: flex;

flex-direction: column;

justify-content: space-around;

align-items: center;

opacity: 0.8;

}

.men-fashion {

background-image: url("../../public/images/men-fashion.jpg");

background-size: cover;

display: flex;

flex-direction: column;

justify-content: space-around;

align-items: center;

opacity: 0.8;

}

.kids-fashion {

background-image: url("../../public/images/kids-fashion.jpg");

background-size: cover;

display: flex;

flex-direction: column;

justify-content: space-around;

align-items: center;

opacity: 0.8;

}

.main-title {

font-family: Arial;

font-size: 50px;

font-weight: bold;

color:white;

text-shadow: 2px 2px 2px black;

}

.btn-shop {

background-color: white;

padding: 10px;

width: 140px;

font-family: Arial;

text-align: center;

box-shadow: 2px 2px 2px black;

border-radius: 5px;

}

.women-fashion:hover, .men-fashion:hover, .kids-fashion:hover {

opacity: 1;

cursor: grab;

}

.services {

display: grid;

grid-template-columns: 3fr 3fr 3fr 3fr;

padding: 20px;

margin-top: 20px;

}

.services .bi-truck, .bi-tag, .bi-lock, .bi-arrow-left-right {

color:red;

}

footer {

background-color: black;

color:white;

font-family: Arial;

padding: 10px;

}

.footer-row-1 {

padding-top: 40px;

padding-bottom: 40px;

text-align: center;

}

.footer-title {

font-size: 30px;

font-weight: bold;

padding-bottom: 30px;

}

.email {

background-color: rgba(135, 131, 131, 0.653);

border:1px solid gray;

padding: 20px;

display: inline-block;

width: 300px;

text-align: left;

}

.subscribe {

background-color: rgb(123, 122, 122);

padding: 20px;

width: 150px;

display: inline-block;

color:white;

margin-left: 10px;

}

.footer-row-2 {

margin-top: 30px;

display: grid;

grid-template-columns: 2.4fr 2.4fr 2.4fr 2.4fr 2.4fr;

padding: 20px;

}

.service-title {

font-size: 16px;

font-weight: bold;

text-transform: uppercase;

display: block;

padding-bottom: 20px;

}

.service-title~span {

display: block;

padding-bottom: 10px;

}

address span {

display: block;

padding-bottom: 10px;

}

aside span {

padding-right: 5px;

}

.footer-brand-title {

font-size: 30px;

font-weight: bold;

padding-bottom: 20px;

display: block;

}

@media screen and (orientation:portrait){

.header-aside, .header-nav{

display: none;

}

.header-brand span:nth-of-type(1n+2) {

display: none;

}

.header-row-2 nav, .short-cuts {

display: none;

}

.bi-justify {

display: inline;

border:none;

font-size: 20px;

}

article {

font-size: 14px;

}

.kids-fashion, .men-fashion {

display: none;

}

main {

display: grid;

grid-template-columns: 12fr;

}

.services {

display: grid;

grid-template-columns: 12fr;

padding: 20px;

margin-top: 20px;

}

.services .bi-truck, .bi-tag, .bi-lock, .bi-arrow-left-right {

color:red;

}

.services div {

padding: 20px;

box-shadow: 1px 1px 2px gray;

margin-bottom: 10px;

}

}

2. shopper-template.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Shopper</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<link rel="stylesheet" href="../src/styles/shopper-template.css">

</head>

<body>

<header>

<div class="header-row-1">

<div class="header-brand">

<span class="bi bi-truck"> FREE SHIPPING WORLDWIDE </span>

<span>United States <span class="bi bi-chevron-down"></span> </span>

<span>USD <span class="bi bi-chevron-down"></span> </span>

<span>English <span class="bi bi-chevron-down"></span> </span>

</div>

<div class="header-nav">

<span>Shipping</span>

<span>FAQ</span>

<span>Contact</span>

</div>

<div class="header-aside">

<aside>

<span class="bi bi-facebook"></span>

<span class="bi bi-twitter"></span>

<span class="bi bi-instagram"></span>

<span class="bi bi-youtube"></span>

</aside>

</div>

<button class="bi bi-justify"></button>

</div>

<div class="header-row-2">

<div>

<span class="brand-title">Shopper.</span>

</div>

<nav>

<span>Home</span>

<span>Catalog</span>

<span>Shop</span>

<span>Blog</span>

<span>Pages</span>

<span>Docs</span>

</nav>

<div class="short-cuts">

<span class="bi bi-search"></span>

<span class="bi bi-person"></span>

<span class="bi bi-heart"></span>

<span class="bi bi-cart4"></span>

</div>

<button class="bi bi-justify"></button>

</div>

</header>

<section>

<article>

<span class="bi bi-lightning-fill"></span>

<span>HAPPY HOLIDAY DEALS ON EVERYTHING</span>

<span class="bi bi-lightning-fill"></span>

</article>

<main>

<div class="women-fashion">

<div class="main-title">Women</div>

<div class="btn-shop">

Shop Women <span class="bi bi-arrow-right"></span>

</div>

</div>

<div class="men-fashion">

<div class="main-title">Men</div>

<div class="btn-shop">

Shop Men <span class="bi bi-arrow-right"></span>

</div>

</div>

<div class="kids-fashion">

<div class="main-title">Kids</div>

<div class="btn-shop">

Shop Kids <span class="bi bi-arrow-right"></span>

</div>

</div>

</main>

<div class="services">

<div>

<span class="bi bi-truck"> </span> FREE SHIPPING

</div>

<div>

<span class="bi bi-arrow-left-right"> </span> FREE RETURNS

</div>

<div>

<span class="bi bi-lock"> </span> SECURE SHOPPING

</div>

<div>

<span class="bi bi-tag"> </span> OVER 10,000 STYLES

</div>

</div>

</section>

<footer>

<div class="footer-row-1">

<div class="footer-title">Want style Ideas and Treats?</div>

<div>

<span class="email">Enter Email\*</span>

<span class="subscribe">Subscribe</span>

</div>

</div>

<div class="footer-row-2">

<div>

<span class="footer-brand-title">Shopper.</span>

<aside>

<span class="bi bi-facebook"></span>

<span class="bi bi-twitter"></span>

<span class="bi bi-instagram"></span>

<span class="bi bi-youtube"></span>

<span class="bi bi-linkedin"></span>

</aside>

</div>

<div>

<span class="service-title">Support</span>

<span>Contact Us</span>

<span>FAQs</span>

<span>Size Guide</span>

<span>Shipping & Returns</span>

</div>

<div>

<span class="service-title">Shop</span>

<span>Men's Shopping</span>

<span>Women's Shopping</span>

<span>Kids' Shopping</span>

<span>Discounts</span>

</div>

<div>

<span class="service-title">Company</span>

<span>Our Story</span>

<span>Careers</span>

<span>Terms & Conditions</span>

<span>Privacy & Cookie policy</span>

</div>

<div>

<span class="service-title">CONTACT</span>

<address>

<span>1-202-555-0105</span>

<span>1-202-555-0106</span>

<span>help@shopper.com</span>

</address>

</div>

</div>

</footer>

</body>

</html>

**CSS Backgrounds**

background-color

background-image

background-size

background-position

background-attachment

background-repeat

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

height: 100vh;

background-image: url("../public/images/a11.jpg"), url("../public/images/a10.jpg");

background-repeat: no-repeat, repeat;

background-size: 600px 300px, 200px 100px;

background-position: center center;

background-attachment: fixed, scroll;

}

</style>

</head>

<body>

<p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Ea suscipit enim deserunt quia vel officia, ratione repudiandae aperiam magnam error sit nam consequuntur voluptas earum pariatur alias amet culpa optio! Lorem ipsum dolor sit amet consectetur, adipisicing elit. Iste iure laudantium magnam autem cumque, unde sequi, dolorem quia, quibusdam voluptates et atque? Atque amet iste quisquam tempore sapiente, dolorem quasi. Lorem ipsum dolor sit amet consectetur adipisicing elit. Officia suscipit, dolor laboriosam temporibus, magnam totam cum ullam eos labore sunt odio, voluptate est numquam excepturi accusantium tempore sint perspiciatis veniam.</p>

</body>

</html>

**CSS Float**

a) **float** : start, end [left, right]

b) **clear** : left, right, both

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

header {

font-family: Arial;

padding: 20px;

display: flex;

justify-content: space-around;

font-size: 20px;

align-items: center;

}

.brand-title {

font-size: 24px;

font-weight: bold;

}

nav span {

padding: 0px 20px 0px 20px;

}

.btn-buy {

background-color: blue;

color:white;

border:none;

padding: 15px;

font-size: 16px;

border-radius: 10px;

}

.img-banner {

width: 50%;

float: right;

}

main {

padding-top: 100px;

padding-left: 100px;

}

.title {

font-size: 50px;

font-family: Arial;

padding-bottom: 20px;

}

.sub-title {

font-size: 22px;

font-family: Arial;

padding-bottom: 40px;

}

.main-title {

color:blue;

}

.btn-doc, .btn-page {

padding: 20px;

width: 200px;

font-size: 20px;

border:none;

}

.btn-page {

background-color: blue;

color:white;

}

.btn-doc {

background-color: white;

border:1px solid gray;

}

</style>

</head>

<body>

<header>

<div>

<span class="brand-title">Landkit.</span>

</div>

<nav>

<span>Landings</span>

<span>Pages</span>

<span>Account</span>

<span>Documentation</span>

</nav>

<button class="btn-buy"> Buy now</button>

</header>

<section>

<img src="../public/images/landkit-banner.jpg" class="img-banner">

<main>

<div class="title">

Welcome to <span class="main-title">Landkit.</span>

Develop anything.

</div>

<div class="sub-title">

Build a beautiful, modern website with flexible Bootstrap components built from scratch.

</div>

<div>

<button class="btn-page">

<span>View all pages </span>

<span class="bi bi-arrow-right"></span>

</button>

<button class="btn-doc">

<span>Documents </span>

<span class="bi bi-arrow-right"></span>

</button>

</div>

</main>

</section>

</body>

</html>

**3/05**

====

Summary

- CSS Integration

- CSS Selectors

- CSS Rules

- CSS Units

- CSS Colors

- CSS Inheritance

- CSS Box Model

- CSS Positions

- CSS Display

- CSS Backgrounds

- CSS Zindex

- CSS Float

**CSS Text Styles**

font-size

font-family

font-weight

font-style

font-variant [small-caps]

text-align

text-decoration

text-shadow

text-indent

text-overflow

overflow

color

white-space

text-transform

letter-spacing

word-spacing

line-height

word-break

**CSS Transforms**

- **Transforming** is the process of converting from one state to another.

- CSS provides

a) 2D Transforms

b) 3D Transforms

- Transformation includes changing position, size, orientation etc.

**2D Transforms:**

- Two dimensional transformations allow to change position, size and orientation along X & Y axis.

- Transform methods include

a) translate()

b) scale()

c) rotate()

d) skew()

e) matrix()

**translate():**

- It allows to change the location of element along X & Y axis.

- It have methods

a) translate() : short hand for both directions

b) translateX() : horizontal

c) translateY() : vertical

Syntax:

{

transform : translate(xPosition, yPosition);

}

Note: Transform is an CSS attribute, which is not supported across all browser.

It requires various plugins to configure for different browsers.

-webkit safari

-moz fire fox

-o opera

-ms Microsoft

Syntax:

{

transform : translateX(300px);

-webkit-transform: translateX(300px);

-moz-transform: translateX(300px);

-o-transform: translateX(300px);

-ms-transform: translateX(300px);

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

img {

transform: translate(0px,0px);

transition: 1s;

}

img:hover {

transform: translate(300px, 100px);

-webkit-transform: translate(300px, 100px);

-moz-transform: translate(300px, 100px);

-o-transform: translate(300px, 100px);

-ms-transform:translate(300px, 100px) ;

transition: 5s;

}

</style>

</head>

<body>

<img src="../public/images/women-fashion.jpg" width="200" height="200">

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.women-fashion {

width: 600px;

height: 400px;

overflow: hidden;

}

.img-women {

width: 605px;

height: 400px;

transition: 100ms;

}

.img-women:hover {

transform: translateX(-5px);

transition: 200ms;

}

</style>

</head>

<body>

<div class="women-fashion">

<img src="../public/images/women-fashion.jpg" class="img-women">

</div>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

.shoe {

width: 250px;

height: 350px;

background-image: url("../public/images/shoe-2.jpg");

background-size: 100%;

display: flex;

flex-direction: column;

justify-content: end;

align-items: center;

overflow: hidden;

}

.shoe:hover {

background-image: url("../public/images/shoe-1.jpg");

background-size: cover;

}

.short-cuts span {

background-color: white;

display: inline-block;

width: 20px;

height: 20px;

border-radius: 20px;

text-align: center;

padding: 10px;

margin: 0px 5px 0px 5px;

font-size: 20px;

}

.short-cuts {

transform: translateY(30px);

transition: 1s;

}

.shoe:hover .short-cuts {

transform: translateY(-10px);

transition: 200ms;

}

.short-cuts span:hover {

background-color: red;

color:white;

cursor: grab;

}

</style>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body>

<div class="shoe">

<div class="short-cuts">

<span class="bi bi-eye-fill"></span>

<span class="bi bi-cart4"></span>

<span class="bi bi-heart"></span>

</div>

</div>

</body>

</html>

**scale()**

- It is used to change the size of element.

- It can transform width using X position & height using Y position.

- Scale methods are

a) scale()

b) scaleX()

c) scaleY()

Syntax:

{

transform : scale(xUnits, yUnits);

}

- Scale units 1=100%

Syntax:

{

transform: scale(1.5, 2); // width=150% height=200%

}

5/05

====

**a) translate()**

**b) scale()**

**c) skew()**

- It is used to tilt element by specified angle.

- It have methods

a) skew()

b) skewX()

c) skewY()

Syntax:

{

transform: skewX(20deg);

}

**d) rotate()**

- It is used to rotate and change the orientation of element.

- You can rotate along X & Y axis by specified angle.

a) rotate()

b) rotateX()

c) rotateY()

Syntax:

{

transform: rotateX(20deg);

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

img {

transform: rotate(0deg);

transition: 1s;

}

img:hover {

transform: rotate(-360deg);

transition: 3s;

}

</style>

</head>

<body>

<img src="../public/images/women-fashion.jpg" width="150" height="200">

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

img {

transform: rotate(0deg) scale(1) skew(0deg);

transition: 1s;

}

img:hover {

transform: rotate(180deg) scale(2) skew(5deg);

transition: 3s;

}

</style>

</head>

<body>

<img src="../public/images/women-fashion.jpg" width="150" height="200">

</body>

</html>

**e) Matrix**

- It allows to apply multiple transforms.

- It can define translate, scale and skew.

Syntax:

{

matrix(scaleX, skewX, skewY, scaleY, translateX, translateY);

}

scale : 1 = 100%

skew : 0.1 = 10deg

translate : 100 = 100px

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

img {

transform: matrix(1,0,0,1,0,0);

transition: 1s;

}

img:hover {

transform: matrix(2,0.2,0,1.5,300,0);

transition: 3s;

}

</style>

</head>

<body>

<img src="../public/images/women-fashion.jpg" width="150" height="200">

</body>

</html>

**CSS 3D Transforms**

- 3D refers to 3 dimensional transformations.

- It includes X, Y & Z axis.

- It have methods

a) translate3D(x,y,z)

b) scale3D()

c) skew3D()

d) rotate3D()

- All 3D transforms require a "perspective", which refers to depth of content.

- 3D transforms require a combination of styles for defining depth.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

img {

transform: scale3d(1,1,1);

transition: 1s;

}

img:hover {

transform: scale3d(2,2,4);

perspective: 50px;

transition: 3s;

box-shadow: 15px 15px 2px black;

}

</style>

</head>

<body>

<img src="../public/images/women-fashion.jpg" width="150" height="200">

</body>

</html>

Task: Design a cube with 5 sides, div with background color or image.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

.box-1 {

width: 200px;

height: 150px;

border:3px solid black;

transform: skewX(-40deg) rotateY(40deg);

background-color: red;

}

.box-2 {

width: 150px;

height: 200px;

border:3px solid black;

background-color: yellow;

transform: translateX(-40px);

}

.box-3 {

width: 150px;

height: 200px;

border:3px solid black;

background-color: blue;

transform: translateX(-40px);

}

</style>

</head>

<body>

<div>

<div class="box box-1"></div>

<div class="box box-2"></div>

<div class="box box-3"></div>

</div>

</body>

</html>

**CSS Transitions**

- Transition controls animation styles for elements.

- It includes

a) transition-duration : total time taken to transform

b) transition-delay : the delay time to start transform.

c) transition-property : defines the property that have to use transition.

d) transition-timing-function : it sets pre-defined animation function.

e) transition [short hand]

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

img {

width: 100px;

height: 100px;

transition-duration: 2s;

}

img:hover {

width: 400px;

height: 400px;

border-radius: 400px;

border: 5px solid red;

transition-duration: 5s;

transition-property: width, height;

transition-timing-function: ease-in;

}

</style>

</head>

<body>

<img src="../public/images/women-fashion.jpg">

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<style>

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

img {

width: 50px;

height: 50px;

transition-duration: 2s;

transition-timing-function: steps(4);

}

img:hover {

width: 500px;

height: 500px;

border-radius: 500px;

border: 5px solid red;

transition-duration: 5s;

transition-timing-function:steps(6);

}

</style>

</head>

<body>

<img src="../public/images/women-fashion.jpg">

</body>

</html>

**CSS Keyframes**

**6/05**

====

**CSS Keyframes**

- Animation comprises of 2 types of frames.

a) Static Frame

b) Key Frame

- Static frame contains content that is same across timeline.

- Keyframe changes the content from start to end.

- Keyframe comprises of

a) Initial State

b) Final State

c) Break Point

- Initial State is defined using "from"

- Final State is defined using "to"

- Break Point can be from 1% to 100%

Syntax:

@keyframes anyName

{

from {

attributes;

}

break%{

attributes;

}

to {

attributes;

}

}

- You can apply keyframe to any element by using animation attributes

a) animation-name

b) animation-duration

c) animation-delay

d) animation-iteration-count

e) animation-timing-function

f) animation-direction

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

@keyframes Blink {

from{

opacity: 0;

transform: scale(0.1);

}

20%{

transform: rotate(180deg);

}

70% {

transform: skew(20deg);

}

to{

opacity: 1;

transform: scale(1);

}

}

.bi-star-fill {

animation-name: Blink;

animation-duration: 1s;

animation-iteration-count: infinite;

}

img {

animation-name: Blink;

animation-duration: 3s;

animation-iteration-count: infinite;

animation-direction: alternate;

}

</style>

</head>

<body style="background-color: black; color:white">

<div>

<div class="bi bi-star-fill" style="font-size: 50px;"></div>

<div><img src="../public/images/women-fashion.jpg" width="50" height="50"> </div>

</div>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

@keyframes Spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

.spinner {

border-left: 15px solid white;

border-right: 15px solid white;

border-top: 15px solid white;

border-bottom: 15px solid red;

width: 100px;

height: 100px;

border-radius: 100px;

animation-name: Spin;

animation-duration: 1s;

animation-iteration-count: infinite;

animation-timing-function: linear;

}

</style>

</head>

<body style="background-color: black; color:white">

<div>

<div class="spinner"></div>

</div>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

@keyframes Spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

img {

animation-name: Spin;

animation-duration: 2s;

animation-iteration-count: infinite;

animation-timing-function: linear;

}

img:active {

animation-duration: 500ms;

}

</style>

</head>

<body>

<div>

<img src="../public/images/fan.png">

</div>

</body>

</html>

https://cssloaders.github.io/

**CSS Media Query**

- It is required to design responsive page.

- CSS media query comprises of specification about media type and condition.

Syntax:

@media type and (condition)

{

}

- "type" refers to screen & print.

- condition is a query that contains attributes

a) width

b) min-width

c) max-width

d) orientation [ landscape, portrait ]

- "width" specifies exactly defined units.

width:600px; exactly at 600px

- "min-width" refers to value starting from specified up to end.

min-width:600px; starting from 600px up to end

- "max-width" refers to value starting from 0 up to specified

max-width:600px; starting from 0 up to 600px

Syntax:

@media screen and (min-width:600px)

{

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<style>

header {

font-size: 20px;

font-family: Arial;

display: flex;

justify-content: space-between;

padding: 20px;

background-color: black;

color: white;

}

button {

background-color: white;

border:none;

}

nav span {

padding-right: 20px;

}

@media screen and (max-width:599px){

nav {

display: none;

}

}

@media screen and (min-width:600px){

nav {

display: inline;

}

button {

display: none;

}

}

</style>

</head>

<body>

<header>

<div>

<span>Shopper.</span>

</div>

<nav>

<span>Home</span>

<span>Shop</span>

<span>Pages</span>

<span>Blog</span>

</nav>

<div>

<button class="bi bi-justify"></button>

</div>

</header>

</body>

</html>

Summary:

- CSS Integration

a) Inline

b) Embedded

c) External File

- Media Type

- MIME Type

- Minification

- CSS Units

- CSS Colors

- CSS Rules

- CSS Selectors

- CSS Box Model

- CSS Positions

- CSS Z-Index

- CSS Float

- CSS Display

- CSS Background

- CSS Text Styles

- CSS List Style

- CSS Transforms

- CSS Transition

- CSS Keyframes

- CSS Animations

- CSS Media Queries

**Bootstrap**

- It is a toolkit used for building responsive and interactive UI.

- You can build fast by using pre-built components, grid-system and layouts.

- Bootstrap up to version 4 require jQuery plugin.

- Bootstrap 5 doesn't require a jQuery plugin, it uses directly JavaScript.

- Bootstrap styles are configured using "Sass" language.

- It is an open source and allows to customize.

- It is cross browser compatible.

**Setup Bootstrap 5:**

> npm install bootstrap bootstrap-icons --save

**Enable Bootstrap:**

<head>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

**7/05**

=====

- Bootstrap provides pre-defined classes for various styles.

- It allows to reuse and customize according to requirements.

**Bootstrap Margins:**

.m-{ } margin all directions {1 to 4}

.mt-{ } top margin

.mb-{ } bottom margin

.my-{ } top and bottom margin

.ms-{ } left margin [start]

.me-{ } right margin [end]

.mx-{ } left & right margin

**Bootstrap Padding:**

.p-{ }

.pt-{ }

.pb-{ }

.py-{ }

.ps-{ }

.pe-{ }

.px-{ }

**Bootstrap Border:**

.border

.border-{size} 1 to 4

.border-{contextual} primary, secondary, success, danger, warning etc.

**Border Radius:**

.rounded

.rounded-{size} 1 to 4

.rounded-pill

.rounded-circle

**Width & Height:**

.w-{ } 25, 50, 75, 100

.h-{ } 25, 50, 100

**Background & Text Color**

.bg-{contextual}

.text-{contextual}

**Text Styles**

.fs-{ } font size 1 to 7

.h{ } heading size from 1 to 6

.fw-bold

.fst-italic

.text-center | end | start | justify

.text-decoration-underline | overline | line-through | none

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body>

<div class="ms-4 mt-4 me-4 rounded rounded-4 border border-3 text-center border-danger p-4 w-25 bg-dark text-warning">

<div class="h1 text-center py-4">Bootstrap</div>

<img src="../public/images/women-fashion.jpg" class="rounded rounded-circle border border-warning border-4" width="200" height="200">

<p class="fs-4 fw-bold text-decoration-underline">Components & Plugins</p>

<dl class="text-start">

<dt>Components</dt>

<dd>Navbar</dd>

<dd>Carousel</dd>

<dd>Alerts</dd>

</dl>

</div>

</body>

</html>

**Bootstrap Display**

.d-none

.d-block

.d-inline

.d-inline-block

.d-flex

.flex-wrap

.flex-row

.flex-column

.justify-content-start | end | center | space-between | space-around | space-evenly

.align-items-start | end | center

Ex:

<div class="d-flex justify-content-center align-items-center" style="height:100vh">

<img src="../public/images/women-fashion.jpg" width="200" height="200" class="border border-2 border-warning rounded rounded-circle">

</div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

<style>

.box {

width: 100px;

height: 100px;

border:1px solid gray;

}

</style>

</head>

<body>

<div class="p-4 m-4 d-flex flex-row justify-content-between flex-wrap">

<div class="box"></div>

<div class="box"></div>

<div class="box"></div>

<div class="box"></div>

<div class="box"></div>

<div class="box"></div>

<div class="box"></div>

<div class="box"></div>

<div class="box"></div>

</div>

</body>

</html>

**Bootstrap Grid**

.row

.col

.col-{size} 1 to 12

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

<style>

.box {

width: 100px;

height: 100px;

border:1px solid gray;

}

</style>

</head>

<body>

<div class="row border m-1 border-4 border-danger p-1">

<div class="col-3 p-4 border border-warning border-2">

col-1

</div>

<div class="col-6 p-4 border border-warning border-2">

col-2

</div>

<div class="col-3 p-4 border border-warning border-2">

col-2

</div>

</div>

</body>

</html>

**Bootstrap Float**

.float-start

.float-end

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="bg-secondary">

<div class="m-4 p-4 bg-white">

<div class="fs-4 fw-bold text-center">Personal Loan EMI Calculator</div>

<div class="row my-4">

<div class="col">

Amount you need <input type="text" size="8">

</div>

<div class="col">

for <input type="text" size="2"> years

</div>

<div class="col">

interest rate <input type="text" size="2"> %

</div>

</div>

<div class="row my-4">

<div class="col">

<input type="range" min="1" max="100" value="20" class="w-100">

<div>

<span>&#8377; 1,00,000/-</span>

<span class="float-end">&#8377; 10,00,000/-</span>

</div>

</div>

<div class="col">

<input type="range" value="2" class="w-100">

<div>

<span>1</span>

<span class="float-end">5</span>

</div>

</div>

<div class="col">

<input type="range" value="10" class="w-100">

<div>

<span>10.45%</span>

<span class="float-end">18.45%</span>

</div>

</div>

</div>

<div class="row">

<div class="col">

<button class="btn btn-primary float-end">Calculate</button>

</div>

</div>

</div>

</body>

</html>

**Bootstrap Position:**

.position-static

.position-absolute

.position-fixed

.position-relative

.position-sticky

.top-{0,50,100}

.end-{0,50,100}

.start-{0,50,100}

.bottom-{0, 50, 100}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body>

<div style="background-image: url('../public/images/women-fashion.jpg'); background-size: cover ; height: 400px;" class="w-50 position-relative">

<div class="bg-danger position-absolute end-0 top-0 rounded rounded-circle text-center text-white p-1" style="width:60px; height: 60px;">20% OFF</div>

</div>

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sapiente saepe in mollitia dolore. Reprehenderit perspiciatis sunt consectetur similique, repellat sint accusamus, cupiditate nostrum iusto praesentium asperiores exercitationem quidem officiis quasi! Lorem ipsum dolor sit amet consectetur adipisicing elit. Rerum quos vel suscipit voluptates exercitationem nobis assumenda perspiciatis quibusdam odio, ut illum nam? Amet quae ipsam iure debitis quod provident non.</p>

<h1 class="bg-dark text-white p-2 position-sticky top-0">News Updates</h1>

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sapiente saepe in mollitia dolore. Reprehenderit perspiciatis sunt consectetur similique, repellat sint accusamus, cupiditate nostrum iusto praesentium asperiores exercitationem quidem officiis quasi! Lorem ipsum dolor sit amet consectetur adipisicing elit. Rerum quos vel suscipit voluptates exercitationem nobis assumenda perspiciatis quibusdam odio, ut illum nam? Amet quae ipsam iure debitis quod provident non.</p>

</body>

</html>

**Button Classes**

.btn

.btn-{contextual}

.btn-{size} sm | lg

.btn-outline-{contextual}

.btn-group | btn-group-vertical

.btn-toolbar

.btn-link

.btn-close

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body>

<dl class="ms-4">

<nav class="btn-toolbar bg-danger mt-4">

<div class="btn-group">

<button class="btn btn-danger bi bi-house-door"> Home</button>

<button class="btn btn-danger bi bi-globe"> Pages</button>

<button class="btn btn-danger bi bi-cart4"> Shop</button>

</div>

</nav>

<dt>Button Base</dt>

<dd>

<button class="btn">Submit</button>

</dd>

<dt>Contextual</dt>

<dd>

<button class="btn btn-primary">Insert</button>

<button class="btn btn-warning">Update</button>

<button class="btn btn-danger">Delete</button>

</dd>

<dt>Outline Contextual</dt>

<dd>

<button class="btn btn-outline-primary">Insert</button>

<button class="btn btn-outline-warning">Update</button>

<button class="btn btn-outline-danger">Delete</button>

</dd>

<dt>Size</dt>

<dd>

<button class="btn btn-outline-primary">Insert</button>

<button class="btn btn-outline-warning btn-sm">Update</button>

<button class="btn btn-outline-danger btn-lg">Delete</button>

</dd>

<dt>Group</dt>

<dd class="btn-group-vertical">

<button class="bi bi-play btn btn-danger"></button>

<button class="bi bi-pause btn btn-danger"></button>

<button class="bi bi-stop btn btn-danger"></button>

</dd>

<dt>Special</dt>

<dd>

<button class="btn btn-link">Signout</button>

<button class="btn btn-close"></button>

</dd>

</dl>

</body>

</html>

**8/05**

====

**Bootstrap Form Classes**

.form-control textbox, password, number, url, email, date etc.

.form-select dropdown

.form-range range input

.form-control-color color input

.form-control-file file input

.form-label label

.form-check-input radio & check box

.form-check-label label for radio & checkbox

.form-switch container that have checkbox or radio

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="d-flex justify-content-center">

<form class="mt-4 border border-2 rounded p-4 w-25">

<div class="bi my-2 text-center bi-person-fill fs-5 fw-bold"> Register User</div>

<div class="mb-2">

<label class="form-label">User Name</label>

<div>

<input type="text" placeholder="Your name" class="form-control">

</div>

</div>

<div class="mb-2">

<label class="form-label">Password</label>

<div>

<input type="password" placeholder="Your password" class="form-control">

</div>

</div>

<div class="mb-2">

<label class="form-label">Your City</label>

<div>

<select class="form-select">

<option>Select City</option>

<option>Delhi</option>

<option>Hyd</option>

</select>

</div>

</div>

<div class="mb-2">

<label class="form-label">Rating</label>

<div>

<input type="range" class="form-range">

</div>

</div>

<div class="mb-2">

<label class="form-label">Subscribe</label>

<div class="form-switch">

<input class="form-check-input" type="checkbox"> <label class="form-check-label"> Yes </label>

</div>

</div>

<div class="mb-2">

<button class="btn btn-warning w-100">Register</button>

</div>

</form>

</body>

</html>

**Bootstrap Input Group**

.input-group

.input-group-sm | lg

.input-group-text

Syntax:

<div class="input-group">

<span class="input-group-text"> </span>

... elements..

</div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body>

<div class="m-4 p-4 w-25">

<div>

<h4>Amazon</h4>

<div class="input-group">

<select class="input-group-text">

<option>All</option>

</select>

<input type="text" class="form-control" placeholder="Search Amazon.in">

<button class="bi bi-search btn btn-warning"></button>

</div>

<h4>Register User</h4>

<div class="mb-4 input-group">

<span class="bi input-group-text bi-person-fill"></span>

<input type="text" class="form-control" placeholder="User Name">

<span class="input-group-text">

<input type="checkbox" class="form-check-input">

</span>

</div>

<div class="input-group">

<span class="bi input-group-text bi-key-fill"></span>

<input type="password" class="form-control" placeholder="Password">

<span class="input-group-text">

<input type="checkbox" class="form-check-input">

</span>

</div>

</div>

</div>

</body>

</html>

**Bootstrap Table Classes**

.table

.table-hover

.table-bordered

.table-borderless

.table-striped

.table-responsive [ bootstrap 5 uses for container that have table ]

.table-dark

.table-{contextual}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="m-4">

<h3>Products Table</h3>

<div class="table-responsive">

<table class="table table-hover table-striped table-bordered">

<thead class="table-dark">

<tr>

<th>Name</th>

<th>Price</th>

<th>Stock</th>

<th>Actions</th>

<th>Name</th>

<th>Price</th>

<th>Stock</th>

<th>Actions</th>

<th>Name</th>

<th>Price</th>

<th>Stock</th>

<th>Actions</th>

<th>Name</th>

<th>Price</th>

<th>Stock</th>

<th>Actions</th>

<th>Name</th>

<th>Price</th>

<th>Stock</th>

<th>Actions</th>

<th>Name</th>

<th>Price</th>

<th>Stock</th>

<th>Actions</th>

</tr>

</thead>

<tbody class="table-success">

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

**Bootstrap Pagination**

.pagination

.page-item

.page-link

.page-item-text

.active

.disabled

.pagination-sm | lg

Syntax:

<ul class="pagination">

<li class="page-item">

<a class="page-link">

<span class="page-item-text"> 1 </span>

</a>

</li>

</ul>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="m-4">

<h3>Products Table</h3>

<div class="table-responsive">

<table class="table table-hover table-striped table-bordered">

<thead class="table-dark">

<tr>

<th>Name</th>

<th>Price</th>

<th>Stock</th>

<th>Actions</th>

</tr>

</thead>

<tbody class="table-success">

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="4">

<ul class="pagination pagination-lg">

<li class="page-item"><a class="page-link"><span class="page-item-text">&laquo;</span></a></li>

<li class="page-item"><a class="page-link"><span class="page-item-text">1</span></a></li>

<li class="page-item"><a class="page-link"><span class="page-item-text">2</span></a></li>

<li class="page-item active"><a class="page-link"><span class="page-item-text">3</span></a></li>

<li class="page-item"><a class="page-link"><span class="page-item-text">4</span></a></li>

<li class="page-item disabled"><a class="page-link"><span class="page-item-text">5</span></a></li>

<li class="page-item"><a class="page-link"><span class="page-item-text">&raquo;</span></a></li>

</ul>

</td>

</tr>

</tfoot>

</table>

</div>

</body>

</html>

**Bootstrap Progress Bar**

.progress-bar

.progress

.progress-bar-animated

.progress-bar-striped

Note: Progress bar value is shown by using style set with "width" in %.

Syntax:

<div class="progress">

<div class="progress-bar" style="width:30%">

</div>

</div>

**Bootstrap Spinners**

.spinner-bordered

.spinner-bordered-sm | lg

.spinner-grow

.spinner-grow-sm | lg

Syntax:

<div class="spinner-bordered text-success"> </div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="m-4">

<h3>Progress</h3>

<div class="progress">

<div class="progress-bar bg-success progress-bar-striped progress-bar-animated" style="width: 30%;">

30% completed

</div>

<div class="progress-bar mx-1 bg-warning progress-bar-striped progress-bar-animated" style="width: 40%;">

40% remaining

</div>

<div class="progress-bar bg-dark progress-bar-striped progress-bar-animated" style="width: 30%;">

30% N/A

</div>

</div>

<h3>Spinners</h3>

<span class="spinner-border text-success"></span>

<button class="btn btn-primary">

<span class="spinner-border spinner-border-sm"></span>

<span>Loading</span>

</button>

<button class="btn btn-primary">

<span class="spinner-grow spinner-grow-sm"></span>

<span>Loading</span>

</button>

</body>

</html>

**Bootstrap Badge**

.badge

.text-{contextual}

.bg-{contextual}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="m-4">

<button class="bi bi-cart4 btn btn-warning position-relative"> <span class="badge bg-danger rounded rounded-circle position-absolute">2</span> </button>

<br><br>

<button class="btn btn-dark"> <span class="badge bg-light text-dark rounded rounded-circle"><span class="bi bi-bell-fill"></span></span> Subscribe </button>

<ul class="">

</ul>

</body>

</html>

Bootstrap List Style

.list-unstyled

.list-group

.list-group-item

.list-group-item-text

.list-group-horizontal

Syntax:

<ul class="list-group">

<li class="list-group-item"> </li>

</ul>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="m-4">

<h2>Table of Contents</h2>

<ul class="list-group">

<li class="list-group-item d-flex justify-content-between list-group-item-danger"><span class="list-group-item-text">Introduction to Web</span> <span class="badge bg-dark rounded rounded-circle">2</span> </li>

<li class="list-group-item d-flex justify-content-between list-group-item-success"><span class="list-group-item-text">Web Terminology</span> <span class="badge bg-dark rounded rounded-circle">5</span></li>

<li class="list-group-item d-flex justify-content-between list-group-item-warning"><span class="list-group-item-text">Web Application Architecture</span> <span class="badge bg-dark rounded rounded-circle">7</span></li>

<li class="list-group-item d-flex justify-content-between list-group-item-info"><span class="list-group-item-text">Servers & Browser</span><span class="badge bg-dark rounded rounded-circle">9</span></li>

</ul>

</body>

</html>

**9/05**

=====

**Bootstrap Cards**

.card

.card-header

.card-body

.card-footer

.card-title

.card-subtitle

.card-img-top

.card-img-bottom

.card-img-overlay

Syntax:

<div class="card">

... card header ...

.... card body....

... card footer...

</div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body>

<main class="m-4 d-flex">

<div class="card p-2 m-2" style="width:300px">

<img src="../public/images/data sci.png" class="card-img-top" height="130">

<div class="card-header">

<h2 class="card-title">Data Science</h2>

<p class="card-subtitle">By Mr. John</p>

</div>

<div class="card-body">

<ul>

<li>Topic-1</li>

<li>Topic-2</li>

<li>Topic-3</li>

</ul>

</div>

<div class="card-footer">

<button class="btn btn-warning w-100"> Join Course </button>

</div>

</div>

<div class="card p-2 m-2" style="width:300px">

<img src="../public/images/power-bi.png" class="card-img-top" height="130">

<div class="card-header">

<h2 class="card-title">Power BI</h2>

<p class="card-subtitle">By Mr. John</p>

</div>

<div class="card-body">

<ul>

<li>Topic-1</li>

<li>Topic-2</li>

<li>Topic-3</li>

</ul>

</div>

<div class="card-footer">

<button class="btn btn-warning w-100"> Join Course </button>

</div>

</div>

</main>

</body>

</html>

**Bootstrap Containers**

- Containers are pre-defined with media queries.

- They are design with various view ports and queries.

.container ] mobile

.container-sm ] mobile

.container-md ] tab

.container-lg ] pc

.container-xl ] laptop

.container-xxl ] laptop wide screen

.container-fluid ] responsive

Syntax:

<body class="container-fluid">

**Bootstrap Components**

- Components are building blocks for UI.

- They enable easy reusability, separation and extensibility.

- A component comprises of pre-defined

a) Design

b) Styles

c) Functionality

**1. Alerts**

- Alert is an embedded message box in page.

- It can dynamically show or hide messages in page.

Classes:

.alert

.alert-dismissible

.alert-{contextual}

.alert-title

.alert-link

Attributes [dynamic]

data-bs-dismiss : It is used to close any component.

It is defined for clickable elements likes buttons.

Syntax:

<div class="alert alert-dismissible">

... your content..

<button data-bs-dismiss="alert"> OK </button>

</div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<h2>Alerts</h2>

<aside class="alert alert-dismissible w-25">

<button class="btn btn-close" data-bs-dismiss="alert"></button>

<div>Ads.</div>

<img src="../public/images/a4.jpg" width="200" height="200">

<p>Offers..</p>

</aside>

<div class="alert alert-success alert-dismissible w-50">

<button data-bs-dismiss="alert" class="btn btn-close"></button>

<h3 class="alert-title">Registered successfully</h3>

<p>Your account created and an activation link sent to your email.</p>

<a class="alert-link" href="#">Help?</a>

<br><br>

<button data-bs-dismiss="alert" class="btn btn-success">OK</button>

</div>

<div class="alert alert-dismissible alert-danger w-25">

<h3>Invalid Credentials</h3>

<p>Please verify your login details</p>

<button class="btn btn-danger" data-bs-dismiss="alert"> OK </button>

</div>

</body>

</html>

**2. Modals**

- Modal is a dialog that pop-up with content in backdrop.

- Modal is hidden and you have to invoke using JS attributes.

- Modal can close using interactive buttons or with backdrop click.

Classes:

.modal

.modal-dialog

.modal-content

.modal-header

.modal-body

.modal-footer

.modal-fullscreen

.modal-dialog-centered

.modal-dialog-scrollable

Attributes

data-bs-target : It refers to the ID of element to open

data-bs-toggle : It define on click to open specific component

data-bs-dismiss : It is used to close component

Syntax:

<div class="modal">

<div class="modal-dialog">

<div class="modal-content">

... header, body, footer

</div>

</div>

</div>

- Every modal must have a reference ID.

- You can access the ID using "data-bs-target".

Note: You can apply ".fade" class to set a fading animation for modal.

Bootstrap 5 doesn't support nested modals.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<button data-bs-toggle="modal" data-bs-target="#login" class="btn btn-primary mt-3">Login</button>

<button data-bs-toggle="modal" data-bs-target="#ads" class="btn btn-danger mt-3">Ads</button>

<div class="modal fade" id="ads">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h3>Summer Sale</h3>

<button class="btn btn-close" data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<img src="../public/images/a4.jpg" width="200" height="200">

</div>

</div>

</div>

</div>

</div>

<div class="modal fade" id="login">

<div class="modal-dialog modal-dialog-centered modal-dialog-scrollable">

<div class="modal-content">

<div class="modal-header">

<h3 class="bi bi-person-circle"> User Login</h3>

<button class="btn btn-close" data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<dl>

<dt>User name</dt>

<dd><input type="text" class="form-control"></dd>

<dt>Password</dt>

<dd><input type="password" class="form-control"></dd>

<dt>User name</dt>

<dd><input type="text" class="form-control"></dd>

<dt>Password</dt>

<dd><input type="password" class="form-control"></dd>

<dt>User name</dt>

<dd><input type="text" class="form-control"></dd>

<dt>Password</dt>

<dd><input type="password" class="form-control"></dd>

<dt>User name</dt>

<dd><input type="text" class="form-control"></dd>

<dt>Password</dt>

<dd><input type="password" class="form-control"></dd>

<dt>User name</dt>

<dd><input type="text" class="form-control"></dd>

<dt>Password</dt>

<dd><input type="password" class="form-control"></dd>

</dl>

</div>

<div class="modal-footer">

<button class="btn btn-primary mx-2">Login</button>

<button class="btn btn-danger" data-bs-dismiss="modal">Cancel</button>

</div>

</div>

</div>

</div>

</body>

</html>

**3. Offcanvas**

- It is similar to modal but supports various orientations.

- It can open with content in backdrop.

Classes:

.offcanvas

.offcanvas-start

.offcanvas-end

.offcanvas-top

.offcanvas-bottom

.offcanvas-header

.offcanvas-body

.offcanvas-title

Attributes:

data-bs-target

data-bs-toggle

data-bs-dismiss

Syntax:

<div class="offcanvas offcanvas-start">

<div class="offcanvas-header">

</div>

<div class="offcanvas-body">

</div>

</div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<header class="mt-4 p-3 border border-2">

<nav class="d-flex justify-content-between">

<button data-bs-target="#navbar" data-bs-toggle="offcanvas" class="bi bi-justify fs-5 btn btn-light"> Shopper </button>

<button data-bs-target="#cart" data-bs-toggle="offcanvas" class="bi bi-cart4 btn btn-warning"></button>

<div class="offcanvas offcanvas-start" id="navbar">

<div class="offcanvas-header">

<h3>Shopping</h3>

<button class="btn btn-close" data-bs-dismiss="offcanvas"></button>

</div>

<div class="offcanvas-body">

<ul class="list-group">

<li class="list-group-item list-group-item-action"> <a>Home</a> </li>

<li class="list-group-item list-group-item-secondary"> <a>Electronics</a> </li>

<li class="list-group-item list-group-item-success"> <a>Fashion</a> </li>

<li class="list-group-item list-group-item-danger"> <a>Footwear</a> </li>

<li class="list-group-item list-group-item-warning"> <a>Contact</a> </li>

</ul>

</div>

</div>

<div class="offcanvas offcanvas-end" id="cart">

<div class="offcanvas-header">

<h3>Your Cart Items</h3>

<button class="btn btn-close" data-bs-dismiss="offcanvas"></button>

</div>

<div class="offcanvas-body">

<div>

<img src="../public/images/a1.jpg" width="100" height="100">

</div>

<div>

<img src="../public/images/a2.jpg" width="100" height="100">

</div>

<div>

<img src="../public/images/a3.jpg" width="100" height="100">

</div>

<div>

<img src="../public/images/a4.jpg" width="100" height="100">

</div>

</div>

</div>

</nav>

</header>

</body>

</html>

**10/05**

=====

**4. Carousel**

- It is used to configure sliding and fading content in page.

- It can be defined with auto and manual control.

Basic Carousel Classes:

.carousel

.carousel-inner

.carousel-item

Syntax:

<div class="carousel">

<div class="carousel-inner">

<div class="carousel-item">

</div>

... multiple items...

</div>

</div>

- Make sure that any one carousel item is active.

<div class="carousel-item active">

</div>

- By default carousel in not defined with animation.

- To start animation you have to define the attribute "data-bs-ride" for carousel.

<div class="carousel" data-bs-ride="carousel">

</div>

- Animation have 2 different effects, you have to define at least any one effect.

a) slide

b) carousel-fade

<div class="carousel slide"> </div>

<div class="carousel carousel-fade"> </div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<div class="carousel slide" data-bs-ride="carousel">

<div class="carousel-inner">

<div class="carousel-item active">

<img src="../public/images/slide-1.jpg" height="250" class="w-100 d-block">

</div>

<div class="carousel-item">

<img src="../public/images/slide-2.jpg" height="250" class="w-100 d-block">

</div>

<div class="carousel-item">

<img src="../public/images/slide-3.jpg" height="250" class="w-100 d-block">

</div>

</div>

</div>

</body>

</html>

**Carousel Controls:**

- Controls allow to navigate between items in a carousel list.

- You can move to previous or next.

Classes:

.carousel-control-prev

.carousel-control-prev-icon

.carousel-control-next

.carousel-control-next-icon

Attributes

data-bs-slide="prev"

data-bs-slide="next"

data-bs-target="carousel\_id"

- You can apply dark theme for controls by using "data-bs-theme=dark".

You have to set theme for carousel.

<div class="carousel" data-bs-theme="dark">

- Bootstrap up to 4 version can use "carousel-dark" class. It is now obsolete.

[Deprecated]

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<div class="carousel slide" data-bs-ride="carousel" id="banners">

<div class="carousel-inner">

<div class="carousel-item active">

<img src="../public/images/slide-1.jpg" height="250" class="w-100 d-block">

</div>

<div class="carousel-item">

<img src="../public/images/slide-2.jpg" height="250" class="w-100 d-block">

</div>

<div class="carousel-item">

<img src="../public/images/slide-3.jpg" height="250" class="w-100 d-block">

</div>

</div>

<button data-bs-target="#banners" data-bs-slide="prev" class="carousel-control-prev">

<span class="carousel-control-prev-icon"></span>

</button>

<button data-bs-target="#banners" data-bs-slide="next" class="carousel-control-next">

<span class="carousel-control-next-icon"></span>

</button>

</div>

</body>

</html>

**Carousel Indicators:**

- Indicators allows random navigation.

- User can navigate to any specific item directly.

Class:

.carousel-indicators

Attributes:

data-bs-slide-to="indexNumber" 0 = first

data-bs-target="carousel\_id"

- Make sure that one indicator button is active.

<button data-bs-slide-to="0" data-bs-target="#banners" class="active"> </button>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<div class="carousel slide" data-bs-ride="carousel" id="banners">

<div class="carousel-inner">

<div class="carousel-item active">

<img src="../public/images/slide-1.jpg" height="250" class="w-100 d-block">

</div>

<div class="carousel-item">

<img src="../public/images/slide-2.jpg" height="250" class="w-100 d-block">

</div>

<div class="carousel-item">

<img src="../public/images/slide-3.jpg" height="250" class="w-100 d-block">

</div>

</div>

<button data-bs-target="#banners" data-bs-slide="prev" class="carousel-control-prev">

<span class="carousel-control-prev-icon"></span>

</button>

<button data-bs-target="#banners" data-bs-slide="next" class="carousel-control-next">

<span class="carousel-control-next-icon"></span>

</button>

<div class="carousel-indicators">

<button class="active" data-bs-slide-to="0" data-bs-target="#banners"></button>

<button data-bs-slide-to="1" data-bs-target="#banners"></button>

<button data-bs-slide-to="2" data-bs-target="#banners"></button>

</div>

</div>

</body>

</html>

**Carousel Timing:**

- The animation of slide and fade have default timing for every item.

- You can set custom timing by using the attribute "data-bs-interval" set with milliSeconds.

Syntax:

<div class="carousel-item" data-bs-interval="4000">

</div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<div class="carousel slide" data-bs-ride="carousel" id="banners">

<div class="carousel-inner">

<div class="carousel-item active" data-bs-interval="3000">

<img src="../public/images/slide-1.jpg" height="250" class="w-100 d-block">

</div>

<div class="carousel-item" data-bs-interval="1000">

<img src="../public/images/slide-2.jpg" height="250" class="w-100 d-block">

</div>

<div class="carousel-item" data-bs-interval="5000">

<div class="d-flex bg-secondary justify-content-center align-items-center" style="height: 250px;">

<div>

<div class="input-group">

<input type="email" placeholder="Your email address" class="form-control">

<button class="btn btn-danger"> Get Started <span class="bi bi-chevron-right"></span> </button>

</div>

</div>

</div>

</div>

<div class="carousel-item" data-bs-interval="4000">

<img src="../public/images/slide-3.jpg" height="250" class="w-100 d-block">

</div>

</div>

<button data-bs-target="#banners" data-bs-slide="prev" class="carousel-control-prev">

<span class="carousel-control-prev-icon"></span>

</button>

<button data-bs-target="#banners" data-bs-slide="next" class="carousel-control-next">

<span class="carousel-control-next-icon"></span>

</button>

<div class="carousel-indicators">

<button class="active" data-bs-slide-to="0" data-bs-target="#banners"></button>

<button data-bs-slide-to="1" data-bs-target="#banners"></button>

<button data-bs-slide-to="2" data-bs-target="#banners"></button>

<button data-bs-slide-to="3" data-bs-target="#banners"></button>

</div>

</div>

</body>

</html>

**5. Collapse**

- It is used to show or hide any container dynamically.

- It requires following classes

.collapse

.show

Attributes:

data-bs-target

data-bs-toggle

Syntax:

<div class="collapse show">

</div>

- Every collapse requires an ID to refer.

- The default collapse state is hidden state

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<div class="mt-4">

<button data-bs-target="#q1" data-bs-toggle="collapse" class="btn btn-dark w-100"> What is Netflix? </button>

<div class="mt-1 collapse" id="q1">

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Est beatae quo aliquam dicta facere perferendis esse rem ab illo inventore perspiciatis incidunt consequuntur voluptatibus, culpa quibusdam explicabo quos. Voluptatibus, facere! Lorem ipsum dolor sit, amet consectetur adipisicing elit. In minima quam mollitia praesentium dolorem quas! A quisquam recusandae amet in quas ratione tempora rerum cupiditate deleniti, eaque harum iste saepe.</p>

</div>

</div>

</body>

</html>

**6. Accordion**

- It is a set of items which you can expand or collapse using Mutex.

- Mutex is mutual exclusion, If any one is expanded then it will collapse all other in the context.

Classes:

.accordion

.accordion-item

.accordion-button

.accordion-header

.accordion-body

.accordion-footer

.accordion-collapse

Attributes

data-bs-target

data-bs-toggle

data-bs-parent [ It is defined for collapse to handle Mutex ]

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<h2 class="text-center">Frequently Asked Questions</h2>

<div class="accordion" id="faqs">

<div class="accordion-item">

<div class="accordion-header">

<button data-bs-target="#q1" data-bs-toggle="collapse" class="accordion-button">What is Netflix?</button>

</div>

<div class="accordion-collapse collapse show" id="q1" data-bs-parent="#faqs">

<div class="accordion-body">

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Non facere voluptatem consequatur, unde ducimus est velit aspernatur, enim repellat pariatur quo odio libero quasi in eligendi! Dolor doloremque sapiente ab? Lorem ipsum dolor, sit amet consectetur adipisicing elit. Praesentium accusamus iure, quos laudantium eligendi dolorem quia asperiores deleniti, quidem doloribus perspiciatis nobis molestias vitae fuga et, quisquam soluta? Atque, aperiam.</p>

</div>

</div>

</div>

<div class="accordion-item">

<div class="accordion-header">

<button data-bs-target="#q2" data-bs-toggle="collapse" class="accordion-button">How much does Netflix cost?</button>

</div>

<div class="accordion-collapse collapse" id="q2" data-bs-parent="#faqs">

<div class="accordion-body">

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Non facere voluptatem consequatur, unde ducimus est velit aspernatur, enim repellat pariatur quo odio libero quasi in eligendi! Dolor doloremque sapiente ab? Lorem ipsum dolor, sit amet consectetur adipisicing elit. Praesentium accusamus iure, quos laudantium eligendi dolorem quia asperiores deleniti</p>

</div>

</div>

</div>

</div>

</body>

</html>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<h2 class="text-center">Frequently Asked Questions</h2>

<div class="accordion" id="faqs">

<div class="accordion-item">

<div class="accordion-header">

<button data-bs-target="#q1" data-bs-toggle="collapse" class="btn btn-dark w-100">What is Netflix?</button>

</div>

<div class="accordion-collapse collapse show" id="q1" data-bs-parent="#faqs">

<div class="accordion-body">

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Non facere voluptatem consequatur, unde ducimus est velit aspernatur, enim repellat pariatur quo odio libero quasi in eligendi! Dolor doloremque sapiente ab? Lorem ipsum dolor, sit amet consectetur adipisicing elit. Praesentium accusamus iure, quos laudantium eligendi dolorem quia asperiores deleniti, quidem doloribus perspiciatis nobis molestias vitae fuga et, quisquam soluta? Atque, aperiam.</p>

</div>

</div>

</div>

<div class="accordion-item">

<div class="accordion-header">

<button data-bs-target="#q2" data-bs-toggle="collapse" class="btn btn-dark w-100">How much does Netflix cost?</button>

</div>

<div class="accordion-collapse collapse" id="q2" data-bs-parent="#faqs">

<div class="accordion-body">

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Non facere voluptatem consequatur, unde ducimus est velit aspernatur, enim repellat pariatur quo odio libero quasi in eligendi! Dolor doloremque sapiente ab? Lorem ipsum dolor, sit amet consectetur adipisicing elit. Praesentium accusamus iure, quos laudantium eligendi dolorem quia asperiores deleniti</p>

</div>

</div>

</div>

</div>

</body>

</html>

**12/05**

=====

**7. Dropdown**

- It can design a dropdown with complex options.

- It supports icons, images and rich format for options.

Classes:

.dropdown

.dropdown-menu

.dropdown-item

.dropdown-item-text

.dropdown-toggle

.dropdown-divider

.dropup

.dropstart

.dropend

Attributes

data-bs-target

data-bs-toggle

data-bs-theme="dark"

Syntax:

<div class="dropdown">

<button class="dropdown-toggle"> Click </button>

<ul class="dropdown-menu">

<li> <a class="dropdown-item"> <span class="dropdown-item-text"> Item </span> </a> </li>

</ul>

</div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<h2>Dropdown</h2>

<div class="dropend mt-4" data-bs-theme="dark">

<button data-bs-toggle="dropdown" class="btn btn-dark dropdown-toggle"> <span class="bi bi-person-circle"></span> Login </button>

<ul class="dropdown-menu">

<li > <a class="dropdown-item" href="../public/shopper-template.html" target="body-frame"> <span class="bi bi-cart4 dropdown-item-text"> Shop </span> </a> </li>

<li > <a class="dropdown-item"> <span class="bi bi-heart dropdown-item-text"> Wishlist </span> </a> </li>

<li > <a class="dropdown-item"> <span class="bi bi-gift dropdown-item-text"> Gift Card </span> </a> </li>

<li > <a class="dropdown-item"> <span class="bi bi-wallet dropdown-item-text"> Payments </span> </a> </li>

<li class="dropdown-divider"></li>

<li > <a class="dropdown-item"> <span class="bi bi-question-circle dropdown-item-text"> Help </span> </a> </li>

</ul>

</div>

<iframe width="100%" height="500" name="body-frame">

</iframe>

</body>

</html>

**8. Nav with Tabs**

- It is a navigation between tabs.

- You can display multiple pages content in one page using tab control.

Classes:

.nav

.nav-tabs

.nav-pills

.nav-menu

.nav-item

.nav-link

.nav-item-text [nav-text]

.tab-content

.tab-pane

Attributes:

data-bs-target [ if your are configuring an anchor elements then href is used]

data-bs-toggle

Syntax:

<ul class="nav nav-tabs">

<li class="nav-item"> <a href="#id" class="nav-link"> </a> </li>

</ul>

<div class="tab-content">

<div class="tab-pane">

</div>

... multiple panes...

</div>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<h2>Product Info</h2>

<ul class="nav nav-tabs">

<li class="nav-item"><a href="#basic" data-bs-toggle="tab" class="nav-link active"> <span class="nav-text">Basic Details</span> </a> </li>

<li class="nav-item"><a href="#preview" data-bs-toggle="tab" class="nav-link"> <span class="nav-text">Preview</span></a></li>

<li class="nav-item"><a href="#offers" data-bs-toggle="tab" class="nav-link"> <span class="nav-text">Offers</span></a></li>

<li class="nav-item"><a href="#reviews" data-bs-toggle="tab" class="nav-link"> <span class="nav-text">Rating & Reviews</span></a> </li>

</ul>

<div class="tab-content mt-4">

<div class="tab-pane active" id="basic">

<h2>iPhone 16 (White) 128 GB</h2>

<div class="fs-2 fw-bold"> &#8377; 69,999</div>

</div>

<div class="tab-pane" id="preview">

<img src="../public/images/iphone-white.jpg" width="200" height="300">

</div>

<div class="tab-pane" id="offers">

<ul>

<li>Offer-1</li>

<li>Offer-2</li>

<li>Offer-3</li>

<li>Offer-4</li>

<li>Offer-5</li>

</ul>

</div>

<div class="tab-pane" id="reviews">

<span class="badge bg-success text-white rounded">4.5 <span class="bi bi-star-fill"></span> </span>

</div>

</div>

</body>

</html>

**9. Navbar**

- It is used to design a responsive navigation bar.

- It can show or hide content using media queries.

- It uses portrait and landscape orientations.

- It also uses min-width and max-width queries.

Classes:

.navbar

.navbar-brand

.navbar-toggler

.navbar-toggler-icon

.navbar-nav

.navbar-collapse

.nav-menu

.nav-item

.nav-link

.nav-text

.navbar-expand-{sm | md | lg | xl }

.navbar-dark | light

.navbar-{contextual}

Syntax:

<nav class="navbar navbar-expand-lg navbar-dark bg-success">

<div class="container-fluid">

<span class="navbar-brand"> Title </span>

<button class="navbar-toggler"> </button>

<div class="navbar-collapse">

<ul class="navbar-nav">

<li class="nav-item"> <a class="nav-link"></a></li>

</ul>

</div>

</div>

</nav>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</head>

<body class="container-fluid">

<nav class="navbar navbar-dark bg-success navbar-expand-lg">

<div class="container-fluid">

<button class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#menu">

<span class="navbar-toggler-icon"></span>

</button>

<span class="navbar-brand">Shopper.</span>

<div class="navbar-collapse collapse" id="menu">

<ul class="navbar-nav">

<li class="nav-item"><a class="nav-link"><span class="nav-text"> Home </span></a></li>

<li class="nav-item"><a class="nav-link"><span class="nav-text"> Shop </span></a></li>

<li class="nav-item"><a class="nav-link"><span class="nav-text"> Docs </span></a></li>

<li class="nav-item"><a class="nav-link"><span class="nav-text"> Pages </span></a></li>

<li class="nav-item"><a class="nav-link"><span class="nav-text"> Blog </span></a></li>

<li class="nav-item"><a class="nav-link"><span class="nav-text">

<div class="input-group">

<input type="text" class="form-control" placeholder="Search Shopper.com">

<button class="bi bi-search btn btn-warning"></button>

</div>

</span></a></li>

</ul>

</div>

</div>

</nav>

</body>

</html>

**10. Breadcrumb**

Classes:

.breadcrumb

.breadcrumb-item

Variable:

--bs-breadcrumb-divider

Syntax:

<ul class="breadcrumb" style="--bs-breadcrumb-divider: ' > ' ">

<li class="breadcrumb-item"> </li>

</ul>

getbootstrap.com => docs => scroll spy

**14/05**

=====

**JavaScript**

- JavaScript is light weight interpreted and JIT compiled programming language.

\* **Light weight** allows to use very less memory.

\* Language is translated by using 2 techniques

a) Interpreted

b) Compiled

\* **Interpreted** allows to translate line-by-line of program.

\* **Compiled** allows to translate all lines simultaneously at the same time.

\* Complied is classified into 2 types

a) JIT

b) AOT

\* **JIT** is Just-In-Time, it compiles only when it is requested.

\* **AOT** is Ahead-of-Time, it is pre-compiled and ready to render.

\* JavaScript is interpreted by using a "JavaScript Interpreter" in browser.

\* JavaScript is compiled using compilers like "Babel, V8, Node etc."

\* A programming enable communication and interaction between use and

application.

- JavaScript is a language that supports various programming paradigms, which

includes structural, functional, imperative, object oriented etc.

\* **Paradigm** provides distinct set of methods, approach and standards to

handle interaction.

\* It uses "Multi Paradigm".

\* Hence it is suitable for various types of technologies.

Note: JavaScript is not an OOP language. It supports only few features of OOP.

- JavaScript is a language which is used in various tiers.

- It is used in

a) Front End

b) Back End

c) Database

- It is used

a) Client Side [ HTML ]

b) Server Side [ Node JS, Next JS]

c) In Database [ MongoDB ]

d) In Action Script [Animation tools]

**JavaScript Client Side:**

- A client side script runs on client device.

- It reduces burden on server, by handling various interactions client side.

- JavaScript client side can handle

a) Browser Interactions [ **BOM** ]

b) Document Interactions [ **DOM** ]

**- Browser Interactions include**

a) window

b) location

c) navigator

d) history etc..

**- DOM interactions include**

a) Data Binding

b) Style Binding

c) Class Binding

d) Event Binding

e) Validations etc.

**Evolution of JavaScript:**

- CERN [ Council for European Research and Nuclear] labs developed a script called ECMA script for Mosaic Browser in early days of Internet.

- In early 1995 Netscape Communications started a browser called "Netscape Navigator".

- Netscape appointed "Brendan Eich" to develop a script for their browser.

- Brendan Eich designed a script by name "Mocha" later renamed as "Live Script".

- Netscape given the responsibility of Live Script to Sun Microsystems. [Java]

- Sun Microsystems renamed the Live Script as "JavaScript".

- Netscape stopped its services in 2000 and given the responsibility of JavaScript to

ECMA.

- ECMA named it as "ECMA Script", Its popular versions are

ECMAScript 2015 ES5

ECMAScript 2016 ES6

ECMAScript 2017 ES7

....

ECMAScript 2024 ES15

**15/05**

=====

**JavaScript with HTML page:**

- JavaScript functions can be defined using following techniques

**1. Inline**

**2. Embedded**

**3. External File**

**Inline:**

- JavaScript functions are defined within HTML element.

- It is faster but not good in reusability.

Syntax:

<button onclick="function(){}"> </button>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<h2>Your Ticket</h2>

<button onclick="window.print()">Print</button>

</body>

</html>

**Embedded:**

- In this technique JavaScript functions are defined in a <script> container.

- You can embed in head or body section.

- It is slower when compared to inline, but allows to reuse.

Syntax:

<script>

function Print()

{

}

</script>

<button onclick="Print()">

- The **MIME** type of script can be "text/javascript", "module", "text/babel" etc.

- If you are using JS in browser to interpret then keep the type as "text/javascript".

Syntax:

<script type="text/javascript">

</script>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script type="text/javascript">

function PrintPage(){

window.print();

}

</script>

</head>

<body>

<h2>Your Ticket</h2>

<button onclick="PrintPage()">Print</button>

</body>

</html>

- JavaScript is not **a strictly typed language**, hence explicitly we have to make it strict.

- **Strict mode** reduces code in-consistency.

- It is defined by using "**use strict**" statement.

Syntax:

<script type="text/javascript">

"use strict";

.... your logic ....

</script>

- JavaScript support 2 types of browser code snippets.

a) **Legacy**

b) **Modern**

- Legacy contains old version related functions.

- Modern have latest version functions.

- If you are writing legacy functions for any older version application, then always

enclose code in HTML comments.

<!-- your comments --> HTML comments

// your comments --> JS Single line comments

/\*

... your comments --> JS Multiline comments

\*/

Syntax:

<script type="text/javascript">

<!--

.... your code ....

-->

</script>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script type="text/javascript">

<!--

"use strict";

function PrintPage(){

window.print();

}

-->

</script>

</head>

<body>

<h2>Your Ticket</h2>

<button onclick="PrintPage()">Print</button>

</body>

</html>

**External File:**

- You can configure JS functions in a script file with extension ".js"

- So that you can access and use across pages.

- However using external file will increase number of requests for page and also page load time.

Syntax:

print.js

function PrintPage(){

window.print();

}

<head>

<script src="print.js"> </script>

</head>

<body>

<button onclick="PrintPage()"> Print </button>

</body>

Ex:

1. src/scripts/print.js

<!--

"use strict";

function PrintPage(){

window.print();

}

-->

2. Home.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script type="text/javascript" src="../src/scripts/print.js">

</script>

</head>

<body>

<h2>Your Ticket</h2>

<button onclick="PrintPage()">Print</button>

</body>

</html>

Note: You have to compress JavaScript file for production.

It requires minification tools.

https://www.toptal.com/developers/javascript-minifier

**FAQ: What are the issues with JavaScript?**

Ans:

1. It is not a strongly typed language.

var x = 10; // x is number

x = "A"; // x is string valid

x = true; // x is Boolean valid

2. It is not implicitly strictly typed language.

3. You need explicit strict mode to turn ON.

4. It is not an OOP language, It supports only few features of OOP.

5. It is not easy to extend.

6. It is not having code level security.

7. It can be blocked by browser.

**FAQ: How to check the status of JavaScript in browser?**

Ans: By using HTML <noscript> element.

Syntax:

<body>

<noscript> please enable JavaScript on your browser </noscript>

</body>

Note: "TypeScript" is an alternative for JavaScript.

However browser can't understand directly typescript.

It is trans compiled into JavaScript.

**16/05**

=====

**Referring HTML elements using JavaScript:**

**1. BOM & DOM hierarchy**

- JavaScript have Browser Object Model & Document Object Model.

- HTML page presents its elements in a DOM hierarchy.

- JavaScript can use the same hierarchy to access elements.

Syntax:

window.document.forms[].elements[]

window.document.images[]

[ ] => refers to a collection of elements starting with index 0.

- DOM reference is native for browser and faster in interaction.

- However referring index number is not good when design changes regularly. As you have to update index number every time when design changed.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function BodyLoad(){

window.document.images[0].src = "../public/images/women-fashion.jpg";

window.document.forms[0].elements[2].value = "Login";

window.document.forms[1].elements[1].value = "Register";

}

</script>

</head>

<body onload="BodyLoad()">

<div>

<img width="100" height="100">

</div>

<div>

<form>

<h3>User Login</h3>

<input type="text" placeholder="User Name">

<input type="password" placeholder="Password">

<input type="button">

</form>

</div>

<div>

<form>

<h3>Register</h3>

<input type="email" placeholder="Your Email">

<input type="button">

</form>

</div>

</body>

</html>

**2. JavaScript can refer HTML elements using a reference name.**

- Every element can have a reference name.

<img name="pic">

<form name="frmHome">

<input name="btnLogin">

- You can access elements by using the name.

pic.src="some\_path";

- You can't access a generic child element without referring to its parent.

buttonName.value="some" ; // invalid

formName.buttonName.value="some"; // valid

- Multiple elements in a page can have same name, this is leads to issues in referring.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function BodyLoad(){

pic.src="../public/images/kids-fashion.jpg";

frmLogin.btnLogin.value = "Login";

frmRegister.btnRegister.value = "Register";

}

</script>

</head>

<body onload="BodyLoad()">

<div>

<img width="100" height="100" name="pic">

</div>

<div>

<form name="frmLogin">

<h3>User Login</h3>

<input type="text" name="txtName" placeholder="User Name">

<input type="password" name="txtPwd" placeholder="Password">

<input type="button" name="btnLogin">

</form>

</div>

<div>

<form name="frmRegister">

<h3>Register</h3>

<input type="email" placeholder="Your Email">

<input type="button" name="btnRegister">

</form>

</div>

</body>

</html>

**3. JavaScript can refer elements by using "ID".**

- Every element can have a reference ID.

<img id="pic">

<input type="button" id="btn">

- You can access elements using ID reference, It requires a DOM method

document.getElementById()

- You can access a child without referring to its parent.

- You can directly access any element.

- ID is used as a selector in CSS, where it can be common for multiple elements.

- This conflicts with JavaScript reference.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function BodyLoad(){

document.getElementById("pic").src = "../public/images/men-fashion.jpg";

document.getElementById("btnLogin").value = "Login";

document.getElementById("btnRegister").value = "Register";

}

</script>

</head>

<body onload="BodyLoad()">

<div>

<img width="100" height="100" id="pic">

</div>

<div>

<form name="frmLogin">

<h3>User Login</h3>

<input type="text" name="txtName" placeholder="User Name">

<input type="password" name="txtPwd" placeholder="Password">

<input type="button" id="btnLogin">

</form>

</div>

<div>

<form name="frmRegister">

<h3>Register</h3>

<input type="email" placeholder="Your Email">

<input type="button" id="btnRegister">

</form>

</div>

</body>

</html>

**4. JavaScript can refer HTML elements using CSS selectors.**

- CSS supports various types of selectors to refer HTML elements.

- JavaScript can directly use CSS selectors with

"document.querySelector()"

- You can access any element directly.

- However same reference name issues will encounter while using selectors.

Syntax:

<img>

<input type="button" id="btn">

document.querySelector("img").src=""

document.querySelector("#btn").value=""

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function BodyLoad(){

document.querySelector("img").src = "../public/images/women-fashion.jpg";

document.querySelector(".btnLogin").value = "Login";

document.querySelector("#btnRegister").value = "Register";

}

</script>

</head>

<body onload="BodyLoad()">

<div>

<img width="100" height="100">

</div>

<div>

<form name="frmLogin">

<h3>User Login</h3>

<input type="text" name="txtName" placeholder="User Name">

<input type="password" name="txtPwd" placeholder="Password">

<input type="button" class="btnLogin">

</form>

</div>

<div>

<form name="frmRegister">

<h3>Register</h3>

<input type="email" placeholder="Your Email">

<input type="button" id="btnRegister">

</form>

</div>

</body>

</html>

**JavaScript Output & Input Techniques**

**17/05**

=====

**Reference Techniques**

- DOM Hierarchy

- Name

- ID

- Query Selector

**JavaScript Output & Input Techniques**

Output Techniques:

1. alert()

2. confirm()

3. document.write()

4. textContent

5. innerHTML

6. innerText

7. outerHTML

8. console methods

**alert():**

- It is a window method.

- It pops up a message box in window.

- You can display any message or result of expression

- It can show static and dynamic values.

- It will not support formats for message. [RC type]

- It will not allow to cancel.

- It is closed only with OK as confirmation.

Syntax:

alert("message | expression");

alert("welcome");

alert( 10 + 20 );

alert( "Addition=" + (10 + 20));

- You can use "\n" for line breaks.

Syntax:

alert("line-1 \n line2 \n line3");

- You can't use any markup as message.

alert("<h2> Welcome </h2>"); // not formatted as heading

- You can't customize by adding, changing or removing elements in alert.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function DeleteClick(){

alert("Delete Record\nRecord Deleted Successfully");

}

</script>

</head>

<body>

<button onclick="DeleteClick()">Delete</button>

</body>

</html>

**confirm():**

- It is similar to alert but allows to cancel.

- It is a Boolean method that returns true or false.

true : on OK click

false : on Cancel click

Syntax:

confirm("message | expression"); true / false

result = confirm(" ");

if(result == true)

{

// actions on OK

}

else

{

// actions on Cancel

}

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function DeleteClick(){

result = confirm("Delete Record\nAre your sure? Want to delete?");

if(result==true){

alert("Deleted Successfully..");

} else {

alert("Delete Canceled..");

}

}

</script>

</head>

<body>

<button onclick="DeleteClick()">Delete</button>

</body>

</html>

**document.write():**

- It is an output method that renders output on a new screen of same page.

- It supports all formats including markup.

- It allows message or expression.

Syntax:

document.write("message | expression | <markup>");

- You can add line breaks using <br> element.

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function DeleteClick(){

result = confirm("Delete Record\nAre your sure? Want to delete?");

if(result==true){

document.write("<h2><font color='red'>Deleted Successfully..</font></h2><br><a href='output.html'>Back</a>");

} else {

alert("Delete Canceled..");

}

}

</script>

</head>

<body>

<button onclick="DeleteClick()">Delete</button>

</body>

</html>

**textContent & innerText :**

- These are the properties used for HTML elements that can display text.

- You can use for containers, semantic or non-semantic.

<header> <nav> <h2> <p> <div> <span> etc.

- They don't support rich formats for text.

- They are RC type. Only plain text is allowed.

- "textContent" is a new option for modern browsers.

Syntax:

document.querySelector("p").textContent = "message | expression";

document.querySelector("div").innerText = "message | expression";

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function DeleteClick(){

result = confirm("Delete Record\nAre your sure? Want to delete?");

if(result==true){

document.querySelector("p").textContent = "Deleted Successfully..";

} else {

document.querySelector("p").innerText = "Delete Canceled";

}

}

</script>

</head>

<body>

<button onclick="DeleteClick()">Delete</button>

<p></p>

</body>

</html>

**innerHTML & outerHTML:**

- They support rich formats.

- You can present using markup.

- innerHTML will display the result inside existing element. [as child]

- outerHTML will display the result by replacing existing element.

Syntax:

document.querySelector("p").innerHTML = "<h2> Welcome </h2>";

<p>

<h2> Welcome </h2>

</p>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function DeleteClick(){

result = confirm("Delete Record\nAre your sure? Want to delete?");

if(result==true){

document.querySelector("p").outerHTML = "<h2>Deleted Successfully..</h2>";

} else {

document.querySelector("p").innerHTML = "<h2>Delete Canceled</h2>";

}

}

</script>

</head>

<body>

<button onclick="DeleteClick()">Delete</button>

<p></p>

</body>

</html>

**console methods:**

- Console is a CLI tool [command line ]

- It is provided in browser debugging tools.

- It is used by developers to test they commands.

- You can use console methods to render output inside console.

- Usually developers use these methods for tracking errors, warnings, info etc.

console.log()

console.warn()

console.error()

console.info()

console.debug()

etc.

- All console methods are RC type.

- They don't support rich formats.

- Line breaks are defined using "\n".

- You can use a message or expression.

Syntax:

console.log("message");

console.warn("expression");

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function DeleteClick(){

console.warn("Delete Clicked");

console.log("Your record will be deteled soon");

result = confirm("Delete Record\nAre your sure? Want to delete?");

if(result==true){

document.querySelector("p").outerHTML = "<h2>Deleted Successfully..</h2>";

console.error("Delete Confirmed");

} else {

document.querySelector("p").innerHTML = "<h2>Delete Canceled</h2>";

}

}

</script>

</head>

<body>

<button onclick="DeleteClick()">Delete</button>

<p></p>

</body>

</html>

**JavaScript Input Techniques**

1. prompt()

2. query string

3. form input elements

**prompt():**

- It is a browser window method.

- It alerts an input box that allow user to input a value.

Syntax:

prompt("your message", "default\_value");

prompt("your message");

- It returns 3 results

a) null on Cancel click with or without value

b) ' ' on OK click without value

c) 'value' on OK click with value

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function SearchClick(){

result = prompt("Enter Search String", "eg: mobiles, watches");

if(result==null) {

alert("You canceled search");

}else if(result=='') {

alert("Please provide a search string\nIt can't be empty");

} else {

document.write("You are searching for : " + result);

}

}

</script>

</head>

<body>

<button onclick="SearchClick()">Search</button>

</body>

</html>

**19/05**

=====

JavaScript Output Techniques

JavaScript Input Techniques

1. prompt()

**2. Query String**

- It is configured in the URL.

- It allows to query any content in page directly from browser address bar.

- It is appended to page using "?".

- It is a key value collection.

**page.html ? key=value**

- You can append multiple keys and values using "&".

**page.html ? key1=value1 & key2=value2 & key3=value3 ...**

- You can access query string using "location.search" property.

location : object

search : property

- JavaScript provides "**URLSearchParams()**" method that can convert query string

into a key and value collection.

**collection = new URLSearchParams(location.search);**

collection.get("key1")

collection.get("key2")

Ex:

1. Create a new page "input.html"

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function bodyload(){

collection = new URLSearchParams(location.search);

document.querySelector("p").innerHTML = "Search Category : " + collection.get('category') + "<br>Searching For :" + collection.get('product');

}

</script>

</head>

<body onload="bodyload()">

<p></p>

</body>

</html>

2. Type the following in address bar after page name

input.html?category=Electronics&product=Mobiles

Ex:

1. Add following pages

- register.html

- result.html

2. register.html uses a form that submits data to result.html

3. result.html uses body load to access and use the query string.

register.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<h2>Register</h2>

<form method="get" action="./result.html">

<dl>

<dt>Name</dt>

<dd><input type="text" name="Name"></dd>

<dt>Price</dt>

<dd><input type="number" name="Price"></dd>

<dt>City</dt>

<dd>

<select name="City">

<option>Select City</option>

<option>Delhi</option>

<option>Hyd</option>

</select>

</dd>

</dl>

<button>Submit</button>

</form>

</body>

</html>

result.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<script>

function bodyload(){

productDetails = new URLSearchParams(location.search);

document.getElementById("lblName").textContent = productDetails.get("Name");

document.getElementById("lblPrice").textContent = productDetails.get("Price");

document.getElementById("lblCity").textContent = productDetails.get("City");

}

</script>

</head>

<body onload="bodyload()">

<h2>Result</h2>

<dl>

<dt>Name</dt>

<dd id="lblName"></dd>

<dt>Price</dt>

<dd id="lblPrice"></dd>

<dt>City</dt>

<dd id="lblCity"></dd>

</dl>

<a href="./register.html">Back to Register</a>

</body>

</html>

**3. Input using Form input elements**

- HTML form provides various input types like text, password, number, email, url, color, file, range etc.

- It also provides elements like select & textarea to input a value.

- JavaScript can use the input elements to access data dynamically from user.

Ex:

src/inox.js

function BookClick(){

document.getElementById("btnContainer").style.display = "none";

document.getElementById("summaryContainer").style.display = "block";

movieName = document.getElementById("lstMovies").value;

document.getElementById("lblMovie").textContent = movieName;

if(movieName=="Mission Impossible"){

document.getElementById("imgPoster").src = "../public/images/mi.jpg";

} else {

document.getElementById("imgPoster").src = "../public/images/final.jpg";

}

document.getElementById("lblCinema").textContent = document.getElementById("lstCinema").value;

document.getElementById("lblDate").textContent = document.getElementById("lstDate").value;

document.getElementById("lblTiming").textContent = document.getElementById("lstTiming").value;

}

function EditClick(){

document.getElementById("lblTitle").textContent = "Modify Booking";

document.getElementById("btnBook").innerHTML = "Save";

document.getElementById("btnBook").className = "btn btn-success";

}

js-examples/inox.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

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

<script src="../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

<script src="../src/scripts/inox.js"></script>

</head>

<body class="container-fluid">

<div id="btnContainer" class="mt-4">

<button data-bs-target="#toolbar" data-bs-toggle="modal" class="btn btn-primary">Quick Booking</button>

</div>

<div class="modal fade" id="toolbar">

<div class="modal-dialog modal-fullscreen">

<div class="modal-content">

<div class="modal-header">

<h3 id="lblTitle">Quick Booking</h3>

<button class="btn btn-close" data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<div class="d-flex justify-content-around">

<div>

<select class="form-select" id="lstMovies">

<option>Select Movie</option>

<option value="Mission Impossible">Mission Impossible:The Final</option>

<option value="Final Destination">Final Destination Bloodlines</option>

</select>

</div>

<div>

<select class="form-select" id="lstDate">

<option>Select Date</option>

<option>Today, 19 May </option>

<option>Tomorrow, 20 May</option>

</select>

</div>

<div>

<select class="form-select" id="lstCinema">

<option>Select Cinema</option>

<option>PVR Panjagutta </option>

<option>Inox B'hills</option>

</select>

</div>

<div>

<select class="form-select" id="lstTiming">

<option>Select Timing</option>

<option> 10:30 AM </option>

<option> 06:30 PM </option>

</select>

</div>

<div>

<button id="btnBook" onclick="BookClick()" data-bs-dismiss="modal" class="btn btn-primary">Book</button>

</div>

</div>

</div>

</div>

</div>

</div>

<div id="summaryContainer" style="display: none;" class="w-25 mt-3 border border-1 rounded p-2">

<h3 class="bi bi-ticket"> Booking Summary</h3>

<img width="100%" height="200" id="imgPoster">

<dl>

<dt>Movie</dt>

<dd id="lblMovie"></dd>

<dt>Date</dt>

<dd id="lblDate"></dd>

<dt>Cinema</dt>

<dd id="lblCinema"></dd>

<dt>Timing</dt>

<dd id="lblTiming"></dd>

</dl>

<button onclick="EditClick()" data-bs-target="#toolbar" data-bs-toggle="modal" class="btn btn-warning bi bi-pen-fill w-100"> Edit Booking </button>

</div>

</body>

</html>

**20/05**

=====

**JavaScript Language**

1. Variables

2. Data Types

3. Operators

4. Statements

5. Functions

**Variables:**

- Variables are storage locations in memory where you can store a value and use it as a part of any expression.

- Variable configuration comprises of

a) Declaration

b) Assignment

c) Initialization

- **Declaration** comprises of a keyword with name.

keyword variable\_name;

Note: Declaring variable is mandatory when JS is in strict mode.

You can ignore declaration if it is not it strict mode.

- Variables in JS can be declared by using following keywords

a) var

b) let

c) const

var variableName;

let variableName;

const variableName;

- **Assignment** is the process of configuring a value into variable after declaration.

var x; // declaring

x=10; // assignment

x=20; // assignment

- **Initialization** is the process of configuring a value into variable while declaring variable.

var x = 10; // initialization

x = 20; // assignment

**FAQ: What is difference between var, let & const?**

**var**:

- It configures a function scope for variable.

- You can declare in any block of a function and use from any another block inside function.

- It supports declaration, assignment and initialization.

- It supports shadowing.

- Shadowing is the process of re-declaring or re-initializing same name identifier again with in the specified scope.

{

var x = 10; // initialization

x = 20; // assignment

var x = 30; // shadowing

}

Ex:

<script>

function f1()

{

var x; // declaring

x = 10; // assigning

if(x==10)

{

var y = 20; // initialization

y = 30; // assignment

y = 40; // assignment

var y = 50; // Shadowing

}

document.write("x=" + x + "<br>y=" + y);

}

f1();

</script>

- It supports hoisting.

- **Hoisting** allows to use and later declare. There is no order dependency of declaring and using a variable.

Syntax:

<script>

x = 10;

document.write("x=" + x);

var x; // hoisting

</script>

**let**:

- It defines a block scope for variable.

- Block scope allows to access within the same block and in its inner block.

- It supports declaration, assignment and initialization.

- It will not support shadowing & hoisting.

Ex:

<script>

"use strict";

function f1()

{

let x; // declaring

x = 10; // assignment

if(x==10)

{

let y = 20;

y = 30; // assignment

document.write("x=" + x + "<br>y=" + y);

}

}

f1();

</script>

**const**:

- It configures a block scope for variable.

- It will allow only initialization.

- It will not allow declaring and assignment.

- It will not allow shadowing and hoisting.

Syntax:

const x ; // invalid

const x = 10; // valid

x = 20; // invalid

**FAQ: Can we change a const ?**

Ans: Yes. While initializing a new value can be sent into constant.

21/05

=====

Variables

- Declaring

- Assignment

- Initialization

- Shadowing

- Hoisting

- Keywords

var, let, const

**Variable Naming:**

- Name must start with an alphabet or underscore "\_" symbol.

- It can be alpha numeric with combination of numbers, but can't start with number.

Syntax:

var sales2025; // valid

var 2025Sales; // invalid

var \_sales; // valid

var $sales; // valid - but not recommended every time.

- The prefix "\_" is used to specify that it is not for external use.

- The prefix "$" is used for library references. [ jQuery ]

- Variable name length can be up to 255 chars. [ Recommended ]

- Name can't be a keyword.

Syntax:

var while; // invalid

var class; // invalid

- Names are case sensitive.

Syntax:

var Form;

var form; // both are different.

var x = 10;

var X = 20;

console.log("x=" + x); // 10

- Name must speak what it is.

Syntax:

var btnInsert; // button for insert

var txtPassword; // text box for password

- Name must be in camel case.

Syntax:

var productPrice;

var studentName;

**Declaration Scenarios:**

1. Continuous set of references

var x, y, z; // all 3 variables use var keyword.

var x; y, z; // y & z are not declared.

var x, y, z = 10; // x=undefined, y=undefined, z=10

var x = y = z = 10; // invalid y and z not declared

var x=10, y=20, z=30; // valid x=10, y=20, z=30

**2. De-structure Declaration**

var [x, y, z] = [10, 20, 30]; // x = 10, y=20, z=30

var [x, y, z] = [10]; // x = 10, y = undefined, z = undefined

var [x, y, z] = 10; // invalid de-structure requires a set of values [ ]

var x, y, z = [10]; // x = undefined, y=undefined, z = array [10]

var x, y, z = [10,20,30]; // z = [10, 20, 30] x=undefined, y=undefined

**Data Types**

- JavaScript is not a strongly typed language.

- It is implicitly typed or dynamically typed.

var x = 10; // x is number type

x = "John"; // x changes to string

x = true; // x changes to Boolean

- Data type is all about the data structure.

- It defines the type of data, range of data and behavior of data.

- JavaScript data types are categorized into 2 groups

1. Primitive Types

2. Non Primitive Types

**Primitive Types**

- They are immutable types.

- They have fixed range for values.

- Value range can't change.

- They are stored in memory stack.

- Stack uses "LIFO" [ Last-In-First-Out ]

- JavaScript Primitive Types are

1. Number

2. String

3. Boolean

4. Null

5. Undefined

6. Big Int

7. Symbol

**JavaScript Number Type:**

- Numeric type in JavaScript can be any one of the following

Signed Integer - 20

Unsigned Integer 20

Floating Point 20.42

Double 450.34

Decimal 3400.34 [ 29 decimal ]

Exponent 2e3 [ 2 x 10^3 ] = 2000

Binary 0b1010

Hexadecimal 0x9311

Octa 0o745

Big Int 938881122n

- Numbers can't be displayed exactly as defined, hence JavaScript provides several methods to display and format numbers.

a) **toFixed()** It sets fixed number of fractions.

b) **toPrecision()** It sets a number to fixed precision of digits.

Syntax:

var x = 4500.34;

x.toFixed(2); // 4500.34

x.toPrecision(5); // 4500.4

x.toFixed(0); // 4500

c) **toLocaleString()** It transforms into a regional string with various formats.

Regional string comprises of formats relative to numbers

currency, fractions etc.

**FAQ: What is localization?**

Ans: Automatically formats numbers, date and time values according to country.

Syntax: Country

var price = 450000;

price.toLocaleString('en-in'); // 4, 50, 000

Syntax: Country with Currency

price.toLocaleString('en-in', { style: "currency", currency: "INR" })

Ex:

<script>

var x = 450000;

document.write("X=" + x.toLocaleString('en-in', {style:"currency", currency:'INR'}));

</script>

Syntax: Local with Percent

var x = 0.785;

x.toLocaleString('en-in', { style: "percent" }); // 79%

Syntax: Local with Unit

var speed = 110;

speed.toLocaleStrign('en-in', { style: "unit", unit : "kilometer-per-hour"}); 110 kph

Syntax: Notation Compact

var views = 1000000;

views.toLocaleString('en-us', { notation: "compact" });

**22/05**

=====

Number Type

- Methods

toFixed()

toPrecision()

toLocaleString()

- style [ currency, percent, unit ]

- notation [ compact ]

- region

**Converting String to Number:**

- JavaScript input techniques return all values in a string format.

- Number input will return a numeric string.

- You can convert numeric string to number by using parsing methods

a) **parseInt()** : for integer values

b) **parseFloat()** : for float, double and decimals

Note: You can convert only a numeric string, which must start with number.

"33ABC" valid

"33AB44" valid

"AB33" invalid

parseInt("33ABC"); // 33

parseInt("33AB44"); // 33

parseInt("AB33"); // NaN

Ex:

<script>

var age = parseFloat(prompt("Enter your age"));

document.write("Your current age is " + age + "<br>");

document.write("You will be " + (age+1) + " next year");

</script>

**Verifying Number:**

- JavaScript can verify numbers by using "isNaN()" method.

- It is a Boolean method that returns true if value is not a number.

Syntax:

if(isNaN("A"))

{

// not a number;

}

Ex:

<script>

var age = parseFloat(prompt("Enter your age"));

if(isNaN(age)){

document.write("Age must be a number");

} else {

document.write("Your current age is " + age + "<br>");

document.write("You will be " + (age+1) + " next year");

}

</script>

**Number Operators:**

- JavaScript provides various operators to handle different types of number interactions.

- Operators include

+ Addition

- Subtraction

\* Multiplication

/ Division

% Modulus Division

\*\* Exponent [ power ]

++ Increment

-- Decrement

Note: The operator "\*\*" is a new from ES5+ version it is power value.

Ex:

<script>

var x = parseInt(prompt("Enter an even number"));

if(x % 2 == 0){

document.write("x=" + x);

} else {

document.write("Only even number allowed");

}

</script>

**Global Scope for Variable:**

- Variables a function or block scope when configured in a function.

- If you want a variable globally accessible to all functions then you can define in

module scope.

- You can use var, let or const for declaring, assigning and initializing global variables.

Syntax:

<script>

// module scope

var x=10;

let y = 20; => global variables

const z = 30;

function f1(){ }

function f2(){ }

</script>

Ex:

<script>

var x = 10;

let y = 20;

const z = 30;

function f1(){

document.write("Function 1 : x=" + x + "y=" + y + "z=" + z + "<br>");

}

function f2(){

document.write("Function 2 : x=" + x + "y=" + y + "z=" + z + "<br>");

}

f1();

f2();

</script>

**- You can configure a global variable inside function by using BOM window object.**

window.name = value;

- It is possible only when JavaScript is used in browser window.

Ex:

<script>

"use strict";

var x = 10;

let y = 20;

const z = 30;

function f1(){

window.a = 70;

document.write("Function 1 : x=" + x + "y=" + y + "z=" + z + "a=" + a + "<br>");

}

function f2(){

document.write("Function 2 : x=" + x + "y=" + y + "z=" + z + "a=" + a + "<br>");

}

f1();

f2();

</script>

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<script>

var volume = 0;

function Increase(){

volume++; // volume = volume + 1

document.getElementById("volume").innerHTML = volume;

}

function Decrease(){

volume--; // volume = volume - 1

document.getElementById("volume").innerHTML = volume;

}

</script>

</head>

<body>

<button onclick="Increase()" class="bi bi-volume-up"></button>

<span id="volume"></span>

<button onclick="Decrease()" class="bi bi-volume-down"></button>

</body>

</html>

**JavaScript Math library:**

- You can't handle all operations directly using operators.

- A Math library is provides to handle complex operations on numbers.

Math.PI

Math.random()

Math.round()

Math.pow()

Math.sin()

Math.cos()

Math.tan()

Math.sqrt()

Math.avg()

Math.floor()

Math.ceil()

etc..

Ex:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

<script>

function GenerateCode(){

var code = Math.round(Math.random()\*10) + "&nbsp;" + Math.round(Math.random()\*10) + "&nbsp;" + Math.round(Math.random()\*10) + "&nbsp;" + Math.round(Math.random()\*10) + "&nbsp;" + Math.round(Math.random()\*10) + "&nbsp;" + Math.round(Math.random()\*10);

document.getElementById("lblCode").innerHTML = code;

}

function RefreshClick(){

GenerateCode();

}

</script>

</head>

<body onload="GenerateCode()">

<dl>

<h3>User Login</h3>

<dt>User Name</dt>

<dd><input type="text"></dd>

<dt>Password</dt>

<dd><input type="password"></dd>

<dt>Verify Code <button onclick="RefreshClick()" class="bi bi-arrow-clockwise"></button> </dt>

<dd id="lblCode"></dd>

</dl>

<button>Login</button>

</body>

</html>