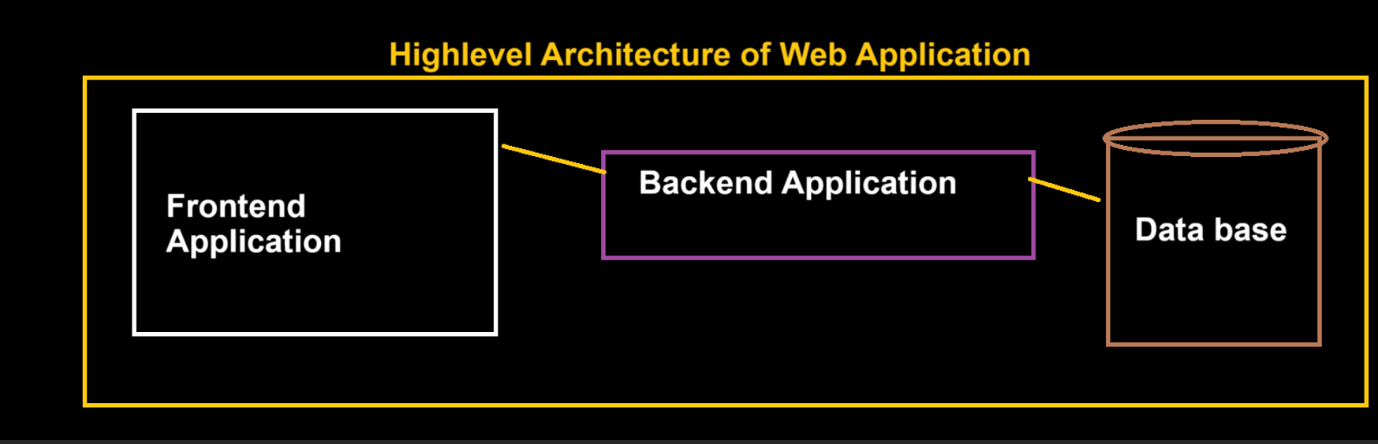
**What is mean by Web Application?**

Any application which is accessed by web is called web application.

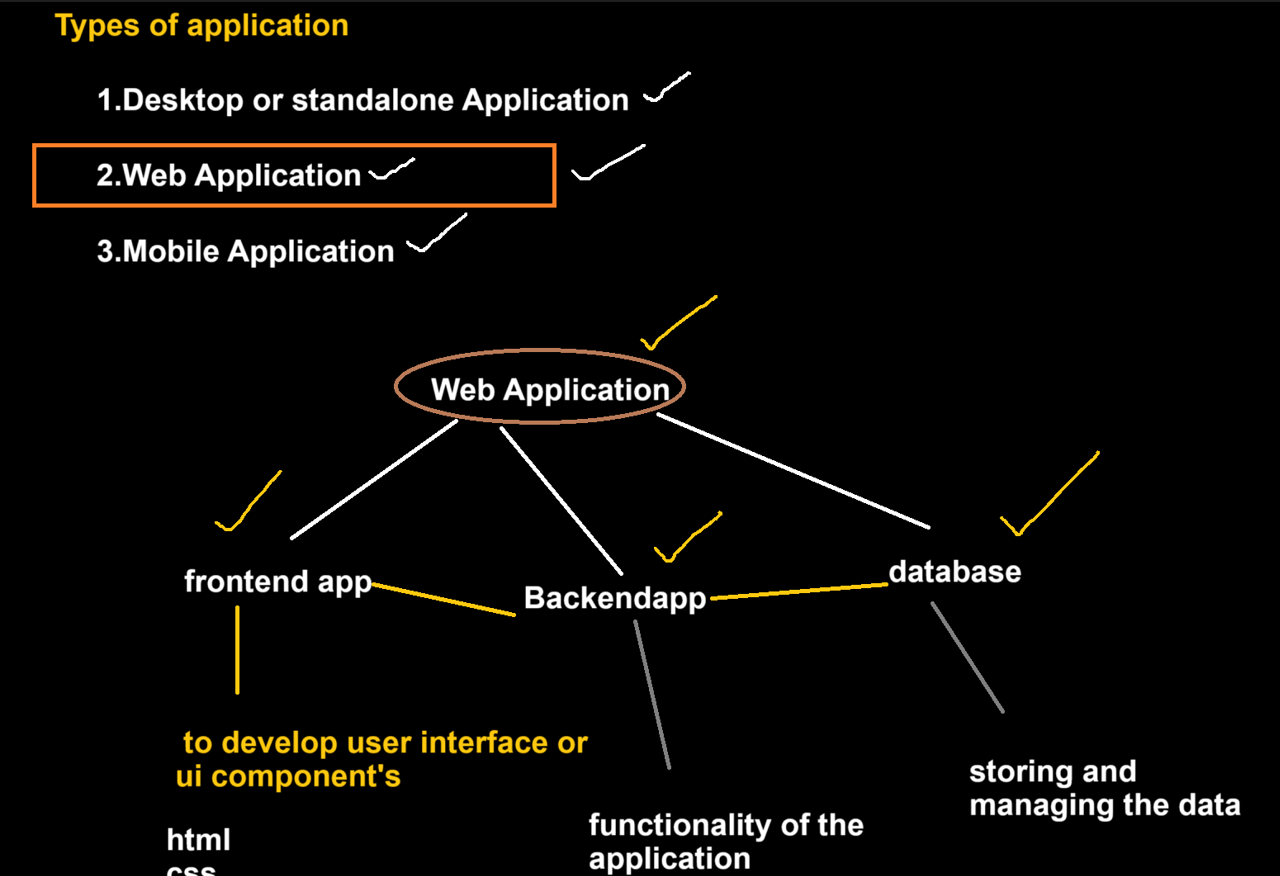


**FRONT END APP or client-side App.**

**BACK-END APP or Server-side App.**

**DATABASE**

**Types Of applications:**



**HTML**

**What is HTML?**

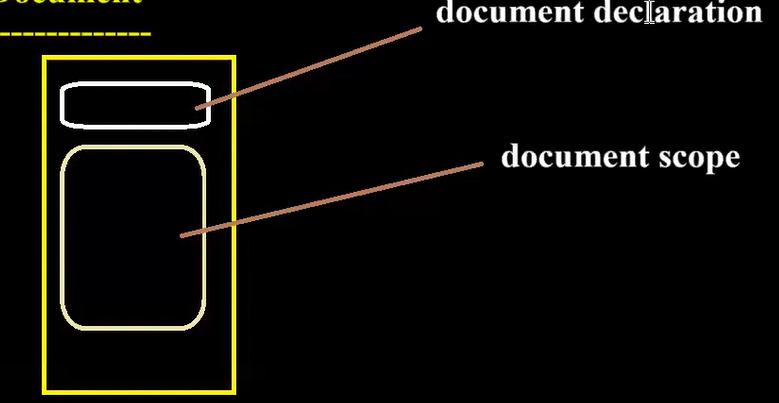
**HTML (HYPER TEXT MARKUP LANG)** is a markup language.

**Why to Learn HTML and its purpose?**

HTML is used to create UI components or User Interface.

**HTML Document Structure**

<!DOCTYPE html>  (Document declaration: version of html )



<html>

<head>

    <title>Document Title</title>  Title for the page

</head>

<body>

    <!-- Content goes here -->

</body>

</html>

**HTML Meta Tags ®**

**○ Charset**

   <meta charset=“utf-8”> specifies page description, keywords, copyright, language, author of the documents, etc.

**○ Name**

**○ Content**

**○ Http-equiv (refresh)**

**HTML Element?**

**HTML Elements are consisting of a start tag, an end tag, and the content between them.**

**HTML Tags**

**○ Container tag (opening and closing tag < > </>)**

**○ Self Tag (only opening tag </>)**

**HTML Attributes**

**“COMMON ATTRIBUTES FOR MOST OF ELEMENTS: (Id, Class, name)”**

**HTML Comments**

**HTML Style Guide**

**HTML Headings**

* **Heading 1 (<h1></h1>)**
* **Heading 2(<h12></h2>)**
* **Heading 3 (<h3></h3>)**
* **Heading 4(<h4></h4>)**
* **Heading 5(<h5></h5>)**
* **Heading 6(<h6></h6>)**

**HTML Paragraphs:**

Using this element, we can add paragraph to the document.

Navigation link can achieve using this anchor element we can connect multiple web pages.

Only one tag that is **“P”** tag, tis paired tag.

**<p> Parah </p>**

**HTML Line Breaks**

**<br/> giving next line.**

**HTML Text Formatting Elements**

* **<b> and <strong> Elements: Bold and strong are like same, for whole parah we can go with strong for particular things we can go with bold**
* **<i> and <em> Elements: Bold and strong are like same, for whole paragraph we can go with strong for particular things we can go with bold**
* **<small> Element: Parah we can go with small for particular things.**
* **<mark> Element: highlight the Parah by default in yellow colour.**
* **<del> Element: Parah will be strike out.**
* **<ins> Element: Parah we can go with small for particular things.**
* **<sub> Element: Parah will be above particular things.**
* **<sup> Element: Parah will be below particular things.**

**HTML Preformatter**

**HTML Links (Anchor Tag)**

Using this element, we can create links

Navigation link can achieve using this anchor element we can connect multiple web pages.

Only one tag that is **“A”** tag, its paired tag

1.We can navigate to from one page to another page. —**INTRA NAVIGATION**

2. We can navigate from one section to other within in this page. **INTER NAVGATION.**

**Attributes of Anchor**

* **Target: where our new page needs to open here whether in same tab or new tab.**

**\_blank: Opens in new tab.**

**\_self: By default, it is self only so no need to mention target for anchor**

**\_parent:**

**\_top: Opens in new tab.**

* **href: The path of HTML page / WEB page where we want to navigate.**
* **id**
* **class**
* **name:**
* **title: (Link will show the title message as hint for the action we are performing)**

**HTML DIV**

Used to create section/layout on the web page.

Used to group multiple elements, it is a plain element.

It provides only one tag called <div> </div> and it is paired tag.

We don’t have any explicit attributes for div elements.

By default no styles will be added.

* **Section:**
* **Layout: It uses rows and columns, which can be achieved using FLEX or GRID from CSS.**

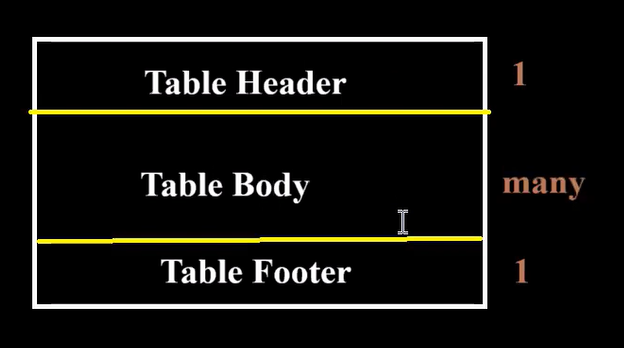
**HTML Images**

**It’s a self-closing tag <img />**

* **Src**
* **Alt**
* **Width**
* **height**

**HTML Image Links**

**HTML Tables**



* **thead**
* **tbody**
* **tfoot**
* **tr**
* **th**
* **td**
* **Colspan**
* **Rowspan**
* **Table fully styles**
* **Nested Table in HTML**

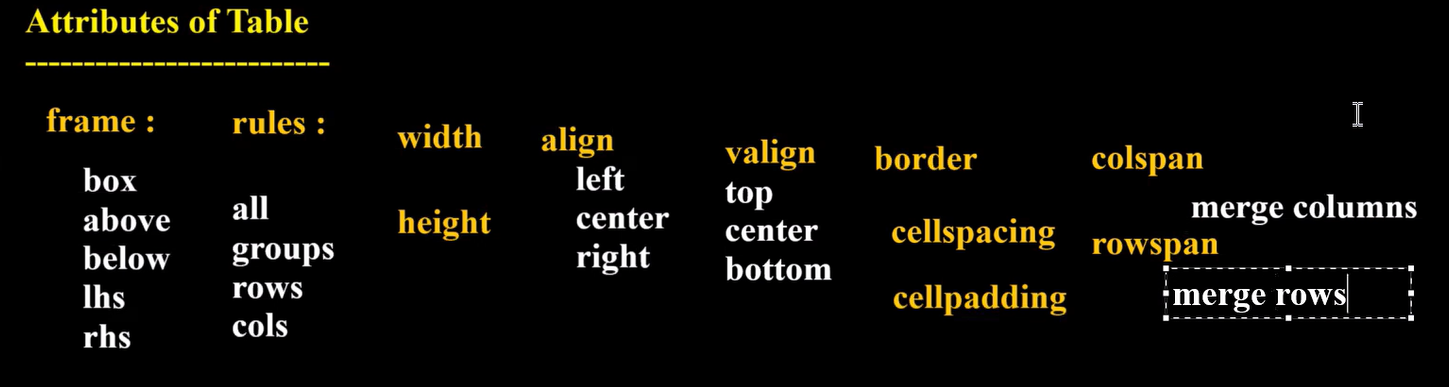
**Attributes of table:**

**Frame:**

**above, below, box, lhs, rhs**

**Rules:**

**all, groups, rows, cols**



**HTML Lists**

* **Unordered HTML List**
* **Unordered Nested**
* **Ordered HTML List**
* **Ordered Nested**
* **Description Lists**

**HTML Block and Inline Elements**

**HTML class Attribute**

**HTML id Attribute**

**HTML Iframes :**

* **it is paired tag**

**HTML Computer Code Elements ®**

* **<code> for Code Elements**
* **<kbd> For Keyboard Input**
* **<samp> For Program Output**

**HTML Semantic Elements / Tags**

* **<header> Element**
* **<footer> Element**
* **<nav> Element**
* **<section> Element**
* **<article> Element**
* **<aside> Element**

**HTML Marquees**

***it is paired tag***

* **Direction**
* **Behavior**
* **Scrolldelay**
* **Scrollamount**
* **Loop etc..**

**HTML Forms**

**It is a paired tag, using form with sub elements can create form in UI.**

**LOGIN form, SIGNIN form, QUIZ form, FEEDBACK form.**

**Form are used to take the data from users.**

* **Form Elements**
* **Form Attributes**
* **Input Types: self-closing tag, can accept lot of attributes.**
* **Placeholder**
* **Text Input Controls**
* **Checkboxes Controls**
* **Radio Box Controls**
* **Select Box Controls**
* **File Select boxes**
* **Hidden Controls**
* **Textarea**
* **Submit and Reset Button**
* **Autocomplete**
* **Required**
* **Disabled**
* **minlength and maxlength etc..**

**HTML New Input Type**

* **color**
* **Number**
* **Range**
* **Date**
* **Datetime-local**
* **Month**
* **Week**
* **Time**
* **Email**
* **Search**
* **Url**
* **Image**

**HTML Progress Tag:**

**Completion progress length**

**HTML Media**

**Video**

* **Source**
* **Autoplay**
* **Controls**
* **Loop**
* **Muted**
* **Src**
* **Poster**

**Audio**



* **Source**
* **Autoplay**
* **Controls**
* **Loop**
* **Muted**
* **Src**

**HTML CSS**



**ELEMENTS PROPERTY**



**TAGS Key:Value**



**UI Components Style to UI component**

**CSS**

**What is CSS?**

**CSS(Cascading Style Sheet) is a styling language.**

**Why to Learn CSS?**

**Why Use CSS?**

**CSS(Cascading Style Sheet) is used to styling the UI component or HTML elements.**

**Advantages of CSS**

**Provides set of or collections of Properties, using this we can style up UI components.**

**What is CSS Syntax**

**CSS Selectors**

* **Simple selectors: (ID, attribute, class, universal)**
* **Combinator selectors**
* **Attribute selectors**
* **Pseudo element selector:**

**STEPS To Add CSS:**

**Select the element which you need to style it, we can select element in two ways.**

**SYNTAX**

**Tag Name or Id Value {**

**CSS PROPERTIES**

**}**

**How To Add CSS**

* **External CSS: reusable style can implement in any document**
* **Internal CSS: style which can be applied within the current document.**
* **Inline CSS: More priority, corresponding element with style property.**

**CSS Colors**

* **RGB Colors**
* **HEX Colors**
* **HSL Colors**

**CSS Backgrounds**

**CSS Fonts Style ®**

* **font-family**
* **font-style**
* **font-weight**
* **font-size**
* **font**

**CSS Text Style**

* **color**
* **direction**
* **letter-spacing**
* **word-spacing**
* **text-align**
* **text-decoration**
* **text-transform**
* **white-space**
* **text-shadow**
* **text-indent**

**CSS Box Model**

**CSS Borders**

**If use pixel it won’t depend on anything.**

**If use % div will depend on body.**

* **border-style**
* **border-width**
* **border-color**
* **border(property)**

**CSS Margins: Space around the element or space outside the border.**

**Values are applied on clock wise direction.**

**Margin: auto (equal space from left and right,exactly center)**

* **margin-top**
* **margin-right**
* **margin-bottom**
* **margin-left**
* **margin**

**CSS Padding: it use to generate space between border and content of the element (or) used to generate space inside the border.**

* **padding-left**
* **padding-right**
* **padding-bottom**
* **padding-top**
* **padding**

**CSS Width**

* **max-width**
* **min-width**

**CSS Height**

* **max-height**
* **min-height**

**CSS Links Style ®**

* **:link**
* **:visited**
* **:hover**
* **:active**

**CSS Outline**

* **outline-width**
* **outline-style**
* **outline-color**
* **outline-offset**
* **Outline**

**CSS Display Property**

* **inline**
* **inline-block**
* **block**

**CSS Float Property**

**CSS Clear Property**

**CSS Cursors Style**

**CSS Overflow Property**

* **overflow-x**
* **overflow-y**
* **overflow**

**CSS Position Property**

* **Static**
* **Absolute**
* **Relative**
* **Fixed**
* **Sticky**

**CSS Z-index Property**

**CSS Combinators ®**

* **descendant selector (space)**
* **child selector (>)**
* **adjacent sibling selector (+)**
* **general sibling selector (~)**

**CSS !important Rule**

**CSS Units ®**

* **(em,rem,vh, vw,vmax,vmin ,px,%)**

**CSS Pseudo Classes ®**

* **:link**
* **:visited**
* **:hover**
* **:active**
* **:focus**
* **:focus-within**
* **:enabled**
* **:disabled**
* **:checked**
* **:required**
* **:read-only / :read-write**
* **:root**
* **:first-child**
* **:last-child**
* **:nth-child()**
* **:nth-of-type()**
* **:first-of-type**
* **:last-of-type**
* **:nth-last-of-type()**
* **:nth-last-child()**
* **:only-of-type**
* **:empty**
* **::first-letter**
* **::first-line**
* **::marker (UL/OL List)**
* **::selection**

**Difference between display none and visibility hidden**

**CSS Rounded Corners**

* **border-top-left-radius**
* **border-top-right-radius**
* **border-bottom-right-radius**
* **border-bottom-left-radius**
* **border-radius**

**CSS Gradients**

**CSS Accent-Color**

**CSS filter Property**

**CSS Media Types (Responsive)**

* **@media print**
* **@media screen**

**CSS Shadow Effects**

* **Text Shadow**
* **Box Shadow**

**CSS 2D Transforms**

* **translate() Method**
* **rotate() Method**
* **scale() Method**
* **skew() Method**

**CSS 3D Transforms**

* **rotateX() Method**
* **rotateY() Method**
* **rotateZ() Method**

**CSS Transitions**

* **transition**
* **transition-delay**
* **transition-duration**
* **transition-property**
* **transition-timing-function**

**CSS Animations**

* **@keyframes**
* **animation-name**
* **animation-duration**
* **animation-delay**
* **animation-iteration-count**
* **animation-direction**
* **animation-timing-function**
* **animation-fill-mode**
* **Animation**

**CSS Flexbox**

* **flex-direction**
* **flex-wrap**
* **flex-flow**
* **justify-content**
* **align-items**
* **align-content ,order**
* **Grid**
* **Grid Template Columns**
* **Grid Template Rows**

**BOOTSTRAP**

**Introduction to Bootstrap? and its Features**

**Bootstrap Installation**

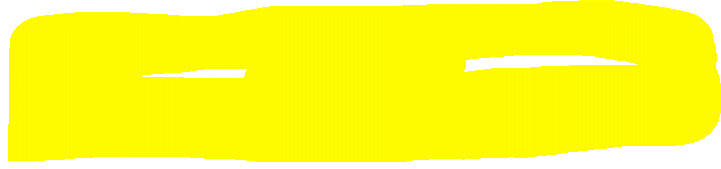
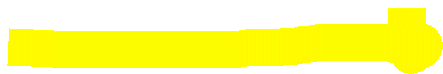
**Setting Development Environment**

**Grid System**

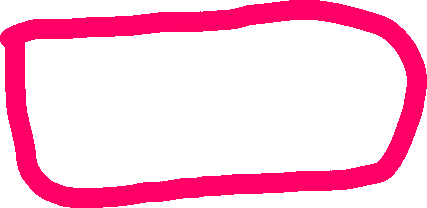
**Components**

* **Basic Typography**
* **Text Alignment & Display**
* **Colors Background**
* **Buttons**
* **Navbars**
* **List groups**
* **Badges**
* **Forms**
* **Carousel**
* **Modal**
* **Table**
* **Cards**
* **Pagination**

**REACT JS:Manage of code is easy**



**Refer another document**



**MONGO DB**

DATABASE: We can store any data and it will store in JSON format.

**INTEGRATION OF MONGODB**

npm install mongoose

**const** *mongoose* = require(**'mongoose'**);

*mongoose*.connect(**'mongodb://localhost:27017/mern-app'**)

.then(()**=>**{

*console*.log(**'DB connected'**)

})

.catch((*err*)**=>**{

*console*.log(*err*);

})

**Connecting to MongoDB Server**

Step 1: Open command prompt(cmd) in your system

Step 2: Type 'mongosh' then hit enter key

**Show database**

show dbs: It will show all the available databases

**Command to connect database**

#Syntax: use [database name]: It will pick you which data base need to work on

|  |  |
| --- | --- |
| How to delete collection:   |  | | --- | | db.collectionname.drop() ex: db.orders.drop() | |

**CRUD Operation**

1. Create
   * insertOne(data,options)
   * insertMany(data,options)

**Creating Collection & Document**

#Syntax: db.collectionName.insertOne({document})

#Example db.products.insertOne({name:'Mac Book Pro',price:120000,brand:'Apple'})

Under the database need to create a collection in that we have to inserted one document.

Example: db.products.insertMany([{name:'Iphone14',price:75000,brand:'Apple'},{name:'Samsung Galaxy A50',price:54000,brand:'Samsung'}])

Under the database need to create a collection in that we have to insert more than one document.

1. Read
   * find(filter,options)
   * findOne(filter,options) #first matching document

**To get all the data from the collection**

#Syntax: db.collectionName.find(): It will show us the document available in that particular collection(bath)

db.customers.find().toArray():All data 0r

db.customers.find().forEach((customersData)=> {printjson(customersData)})

db.products.find({},{name:1,price:1}) :it will list only name and price along with id

db.products.find({},{\_id:0,name:1,price:1}) :it will list only name and price.

#Syntax: db.collectionName.find({particular document}): It will show us current document available in that particular db

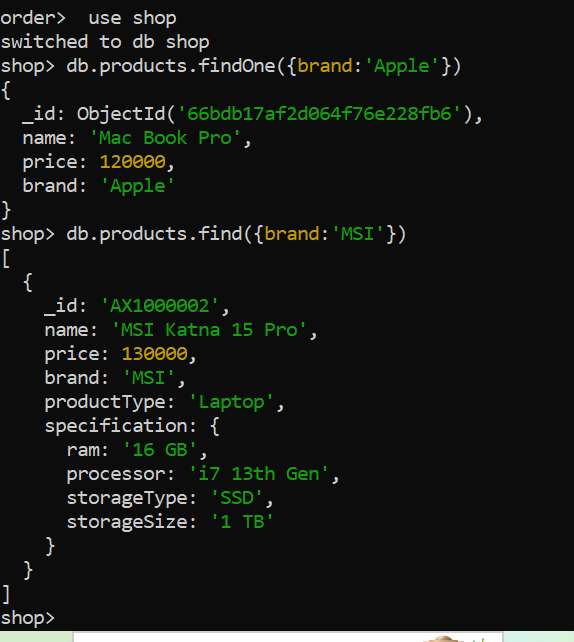
#Syntax: db.collectionName.findOne({particular document}): It will show us first one of the current document available in that particular db

Example:

db.products.find({brand:'Apple'}) :

Example:

db.products.findOne({brand:'Apple'})



1. Update
   * updateOne(filter,data,options)
   * updateMany(filter,data,options)
   * replaceOne(filter,data,options)

**Update document in collection**

#Syntax: Db.collectionName.updateOne({},{}): filter and what to set

**Example:**

db.products.updateOne({name:'Mac Book Pro'},{$set:{productType:'Laptop'}})

**Example:**

db.products.updateOne({name:'Samsung Galaxy s24 Ultra'},{$set:{price:93000}})

**Example:**

db.products.updateOne({name:'Bravia 43 inch 4k'},{$set:{price:45000}})

**Example:**

db.products.updateOne({name:'Iphone 15'},{$set:{brand:'Apple',productType:'Mobile'}})

**Example:**

db.products.updateMany({\_id:{$in:[ObjectId('66695c4aa52b865509cdcdf8'),ObjectId('66695c8ba52b865509cdcdfd')]}},{$set:{productType:"Mobile"}})

**Example update many**

db.products.updateMany({\_id:{$in:[ObjectId('66695c4aa52b865509cdcdf8'),ObjectId('66695c8ba52b865509cdcdfd')]}},{$set:{productType:"Mobile"}})

**Replacing of entire document:**

db.products.replaceOne({\_id:ObjectId('66695c8ba52b865509cdcdfc')},{name:"Iphone 14 Pro Max",price:115000,brand:'Apple',productType:'Mobile'})

1. Delete
   * deleteOne(filter,options)
   * deleteMany(filter,options)

## Delete document from collection

# the following command will delete the 1st document with matched filter

db.products.deleteOne({brand:'Apple'})

# the following command will delete document many at same time

db.products.deleteMany({productType:'Laptop'}) documents matches the filter will delete from collections

db.products.deleteMany({})--- all documents will delete from collections

**deleting collection:**

db.products.drop()---collection will delete

**Deleting Database:**

db.dropDatabase()

**EXPRESS JS**

It is framework for NODE JS

npm install express.

In package.json(type:module)-🡪

import *express* from **"express"**;

**const** *app* =  express();

(OR)

const express = require('express');

const app = express();

**NODE JS**

It is javascript run time, if want to run javascript outside of browser NODE JS is required.

NODE JS have **v8 Engine** it will support javascript performance related things.