CCEE react component state

shortcut keys

extension es7\_ react.redux.react

rfc shortcut for easy component

types of components for sharing data : identigy the relation between the components

parent - > child component (App - > Login) --> props(properties)

child component - parent - >(Login - > App) --> callBack Method BMIFORM-APP

sibling components ---- context

data needs to be shared among multiple elements in a hierarchy -- redux --

state management tool available in JS

var is kind of global scope variable

var x = "a";

var x = "b";

multiple declaration ⬆️

let

arrow function\

left side input and right side output

[bmi,setBmi] = useState(18);

we are declaring a variable bmi

ES6 -->

create a google page

create a product page

create colourful resume

-->> DESTRUCTURING

-> Import / Export

-> Passing parameter to function (UNORDERED)

function printEmailAndNameV1(user){

console.log(user.name);

console.log(user.email);

}

//Index Based Ordered Param

function printEmailAndNameV2(name,email){

console.log(user.name);

console.log(user.email);

}

/

function printEmailAndNameV3({name,email}){

console.log(user.name);

console.log(user.email);

}

let user = {id:!,name:"abc",email:"abc@gmail.com",address:"mumbai"};

printEmailAndNameV1(user);

printEmailAndNameV2(user.name,user.email);

printEmailAndNameV3(user);

==================================================================================

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

let user = {id:!,name:"abc",email:"abc@gmail.com",address:"mumbai"};

// conventional programming

let name = user.name;

let email = user.email;

let address = user.address

// Adv JS

let {name,email} = user;

console.log(name,email);

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

==================================================================================

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

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

DESTRUCTURING OF OBJECT

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

let user = {

"userId": 1,

"id": 1,

"title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",

"body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"

}

let {title,body} = user;

let {body,title} = user; // {}-> UNORDERED

console.log(title);

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

==================================================================================

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

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

DESTRUCTURING OF OBJECT

list(dynamic array)

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

let list = ["mumbai","pune","nagpur","banglore","tvm","kochi","calcultta"]

///////////////////////////

// conventional programming

let city1 = list[0];

let city2 = list[1];

let city3 = list[2];

console.log(city1);

console.log(city2);

console.log(city3);

///////////////////////////

// First three / top 3 city

let [city1,city2,city3]=list;

// access the second element

let [ , city2]=list;

// access the second element

let [ , p2,p3]=list;

console.log(city1);

console.log(city2);

console.log(city3);

console.log(p2,p3);

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

==================================================================================

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

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

SPREAD OPERATOR :: COPY :: CLONE

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

let user1 = { id:1,name:"abc",

assignment1:"react assignment done",

assignment2:"react assignment done",

assignment3:"react assignment done"};

// Clone :: SPREAD OPERATOR(...)

// example -- 1

let user2 = { ...user1 };

{ ...user1 } ==>> {id : user1.id, name : user1.name, assignment : user1.assignment, ......};

console.log(user2);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// example -- 2

// REACT + VIRTUAL DOM

// Clone + override name and id

let user2 = { ...user1,name:"def",id:2};

console.log(user2);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// example -- 3

// REACT + VIRTUAL DOM

// Clone List

let list = ["mumbai","pune","nagpur","banglore","tvm","kochi","calcultta"];

let list1 = [...list1];

[...list1] --->[list[0],list[1],list[2],list[3],list[4],list[5],list[6]];

console.log(list1);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pls check timestamp 10:03 -->23-05-2023

// example -- 4

// REACT + VIRTUAL DOM

// Clone List

let ref1 = {};

// This is error , ref1 is object literal, ref2 is array

let ref2 = [...ref1]

let list = []

let ref3 = {..ref1}

// Both will give error

⬇️⬇️⬇️⬇️⬇️⬇️⬇️

--> cloning object will get object only similiar

--> cloning array will get array only

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

==================================================================================

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

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

MAP OPERATOR :: LOOPING IN REACT

THIS WORKS WITH ARRAY OR LIST

THIS IS EQUIVALENT TO THE LAMBDA AND STREAM IN JAVA

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// example -- 1

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

// APPROACH 1

// INPUT

let list = ["mumbai","pune","nagpur","banglore","tvm","kochi","calcultta"];

// OUTPUT

// ["Hi MUMBAI","Hi PUNE","Hi NAGPUR","Hi BANGLORE","Hi TVM","Hi KOCHI","Hi CALCUTTA"]

let list1=[];

for(let item of list){

let str= "Hi " + item.toUpperCase();

list1.push(str);

}

console.log(list);

console.log(list1);

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

// APPROACH 2

// INPUT

let list = ["mumbai","pune","nagpur","banglore","tvm","kochi","calcultta"];

// OUTPUT

// ["Hi MUMBAI","Hi PUNE","Hi NAGPUR","Hi BANGLORE","Hi TVM","Hi KOCHI","Hi CALCUTTA"]

let list2 = list.map((item))=> "Hi" +item.toUpperCase();

console.log(list);

console.log(list2);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// example -- 2

let list = ["mumbai","pune","nagpur","banglore","tvm","kochi","calcultta"];

// <h1>mumbai</h1>,<h1>pune</h1>,<h1>nagpur</h1>,<h1>banglore</h1>,<h1>tvm</h1>,

<h1>kochi</h1>,<h1>calcutta</h1>

let list1 = list.map((item)=>`<h1>${item}</h1>`);

console.log(list1);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

display timestamp in messaging html

backtick or backspace we can write multiline statement

`<div fred

ferw

fer

asretgsfvd

gterv

fegrtvfs>grefvsdgrtfv </div>`

in bootstrap we can have 12 columns per row

GET VS POST

GET is default HTTP method.

GET is transparent method.

In this data is visible in URL

Limitations

1.GET should not be used for SENSITIVE DATA

2.Max length of URL ? 2024

3.Binary File Restrictions

POST

POST method is mainly use for sending senstitive data to server.Is not is visible to URL

No Maxlength

Users can upload images and send binary data

It hides data

Different type of for loop for getData

// for (let item of data) {

// let newElement = `

// <div class="bg-success text-light p-2 rounded fs-6 mb-1">

// ${JSON.stringify(item)}

// </div>`;

// parent.innerHTML += newElement;

// }

// for (let item of data) {

// let { \_id, ...doc } = item; // Exclude the "\_id" field from the document

// let newElement =

// "<div class='bg-success text-light p-2 rounded fs-6 mb-1'>";

// // Iterate over each key-value pair in the document

// for (let [key, value] of Object.entries(doc)) {

// newElement += `<p><strong>${key}:</strong> ${value}</p>`;

// }

// newElement += "</div>";

// parent.innerHTML += newElement;

// }

// for (let item of data) {

// let { \_id, ...doc } = item; // Exclude the "\_id" field from the document

// let newElement =

// "<div class='bg-success text-light p-2 rounded fs-6 mb-1'>";

// for (let key in doc) {

// newElement += `<p><strong>${key}:</strong> ${doc[key]}</p>`;

// }

// newElement += "</div>";

// parent.innerHTML += newElement;

// }

// data.forEach((doc) => {

// let newElement = document.createElement("pre");

// newElement.textContent = JSON.stringify(doc, null, 2);

// parent.appendChild(newElement);

// });

<Header></Header>

<hr></hr>

<MainBody />

<hr></hr>

<Footer></Footer>

mongodb://127.0.0.1:27017

//time stamp 12:00 to 12:30//

no curly brackets means external default module

express help build a server between client

/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\

\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/

import {MongoClient,ObejctId} from "mongodb";

import express from "express";

// inititalizing the express

const app = express();

//middleware that helps body to json

app.use(express.json); // for parsing application json

app.use(express.urlencoded({})); // for parsing applicatiox-www-form-urlencoded

//whitelisting all request

app.use(cors());

[

localhost:4000 domain ajax request call

for accessing from different domain

npm install cors

]

//handling the client request

// api will look like this

// http://localhost:4000/

// http://127.0.0.1:4000/

// GET => get the data from the server

// anonymous function / remote function

app.get("/",function(req,res){

res.json({});

});

// http://localhost:4000/users

//another api

app.get("/users/",(req,res)=>{

let list = [];

let user1 = {id :1,name:"abc"};

let user2 = {id :2,name:"def"};

list.push(user1);

list.push(user2);

res.json({});

});

// static get method

// http://localhost:4000/users/1

// http://localhost:4000/users/1

app.get("/users/1/",(req,res)=>{

let user = {id :3,name:"ghi"};

});

// http:// localhost:4000 / users / : id

// dynamic get method

app.get("/users/:id",(req,res)=>{

let user = {id :3,name:"ghi"};

});

//dynamic

app.get("/users/1/",(req,res)=>{

let user = {id :req.params.id , name:"ghi"};

});

// http://localhost:4000/posts

app.get("/",(req,res)={

let username = req.query.username;

let password = req.query.password;

res.json({username:username.password:password});

});

// POST => insert/create/add new data to server

// semantic meaning post new data to server

// http://localhost:4000/user

//remote method

app.post("/user/",(req,res)=> {

// logic

// db logic

console.log(req.body);

res.json({msg : "im post"})

});

// create multiple post api

// CURD semantic meaning of protocol

// C - POST

// U - PUT

// R - GET

// D - DELETE

(in case of fetch only particular value)

//setting the server at port 4000

// starting the server

app.listen(4000);

[][][][][][][][][][]][][]][][][]][][][][][[][][][][]

SEMANTIC meaning

// GET

// GET ME ALL THE USERs

app.get("/users");

app.get("/users/",async (req,res)=>{

const uri = "mongodb://localhost:27017";

const client = new MongoClient(uri);

let db = client.db("database name");

let list = await db.collection("user".find().toArray());

await client.close();

});

// GET

// GET ME USER WHERE ID IS PARAM

app.get("/users/:id");

app.get("/users/:id",async (req,res)=>{

const uri = "mongodb://localhost:27017";

const client = new MongoClient(uri);

let db = client.db("database name");

let id = new ObjectId(req.params.id);

let list = await db.collection("user".find(id));

await client.close();

});

// POST

// CREATE NEW USER

// http://localhost:4000/user

app.post("/user");

app.post("/user",async (req,res)=>{

const uri = "mongodb://localhost:27017";

const client = new MongoClient(uri);

let userDoc = req.body;

const db = client.db("database name");

await db.collection("user".insertOne(userDoc));

await client.close();

res.json({por.true});

});

[][][][][][][][][][][][][][][][][][][][][][][][][][][][][]

--------------------------Sir CODE------------------------

[][][][][][][][][][][][][][][][][][][][][][][][][][][][][]

// GET

// GET ME ALL THE USERs

// http://localhost:4000/users

app.get("/users", async (req, res) => {

const uri = "mongodb://localhost:27017";

const client = new MongoClient(uri);

let db = client.db("project");

let list = await db.collection("user").find().toArray();

await client.close();

res.json(list);

});

// GET

// GET ME USER WHER ID IS PARAM

app.get("/users/:id", async (req, res) => {

const uri = "mongodb://localhost:27017";

const client = new MongoClient(uri);

let db = client.db("project");

let id = new ObjectId(req.params.id);

let user = await db.collection("user").findOne({ \_id: id });

await client.close();

res.json(user);

});

app.get("/users/query\_by\_username/:username", async (req, res) => {

const uri = "mongodb://localhost:27017";

const client = new MongoClient(uri);

let db = client.db("project");

let user = await db

.collection("user")

.findOne({ username: req.params.username });

await client.close();

res.json(user);

});

// POST

// Create New User

// http://localhost:4000/user

app.post("/user/", async (req, res) => {

const uri = "mongodb://localhost:27017";

const client = new MongoClient(uri);

let userDocument = req.body;

const db = client.db("project");

await db.collection("user").insertOne(userDocument);

await client.close();

res.json({ opr: true });

});

// DELETE

// DELETE ONE USER

// http://localhost:4000/user/646866fe5d8996044b96ce6c

app.delete("/user/:id", async (req, res) => {

const uri = "mongodb://localhost:27017";

const client = new MongoClient(uri);

const db = client.db("project");

let id = new ObjectId(req.params.id);

await db.collection("user").deleteOne({ \_id: id });

await client.close();

res.json({ opr: true });

});

[][][][][][][][][][][][][][][][][][][][][][][][][][][][][]

--------------------------Sir CODE------------------------

[][][][][][][][][][][][][][][][][][][][][][][][][][][][][]

// DELETE

// DELETE ONE USER

// http://localhost:4000/user

app.post("/user/:id");

[ BY DEFAULT FETCH CALLS GET METHOD = > fetch(url)]

.gitignore use for hiding file/folder

i.e., is not part of git

in this file will include the name of files folder to hide/ignore

main file to hide node\_modules

with the help of export we can make the function(s) in js is public

export{func1,func2};

// approach 1

export{func1};

export{func1};

// approach 2

export func1(){

}

export func2(){

}

export default func4(){

}

WE CAN MAKE ONLY ONE FUCTION AS DEFAULT

when the function is default name doesn't matter

now importing in another in js file

import{func1,func2} from "./pattern.js";

here "./" is means local module

another approach

import{func1} from "./pattern.js";

import{func2} from "./pattern.js";

import func4 from "./pattern.js";

function func3(){

console.log("ge");

func1();

}

before that mention "type":"module" in package.json

cannnot use await without declaring function async

npm i react

js file name should be correct if there path include path in src too

inline js also runnable

with javascript makes application interactive

html is skeleton

DOM

CSS SELECTOR

-NAME

-ID (#)

-CLASS (.)

document :: Table

SELECT \* FROM TABLE; :: document

SELECT \* FROM TABLE WHERE ID=1 :: document.querySelector("#id")

// ELEMENT SELECTOR

document.querySelector("h1");

// ID SELECTOR

document.querySelector("#id1");

# CLASS SELECTOR

document.querySelector(".class1");

object literals

function f1(){

let obj = {};

let obj2 = {id:1,name:"Tej"};

// modify member

obj2.id=12;

// using bracket

obj2["name"] = "Tejas";

// add member

obj2.email="abc@.com";

// delete member

obj2.email;

console.log(obj);

console.log(obj2.id);

console.log(obj2.name);

}

JSON

JSON is data transfer language

object creation

function f1(){

let obj = {};

let obj2 = {"id":1,"name":"Tej"};

let obj2 = `{"id":1,"name":"Tej"}`; // this is json string

let obj

console.log(obj);

console.log(obj2.id);

console.log(obj2.name);

// PURPOSE

// JSON STRING TO OBJECT LITERALS

let jsonStr = `{ "id": 1, "name": "rohit", "email": "rohit@gmail.com" }`;

let obj = JSON.parse(jsonStr);

console.log(typeof jsonStr);

console.log(typeof obj);

// OBJECT LITERAL TO JSON STRING

let str1 = JSON.stringify(obj);

console.log(str1);

}

function main() {

let obj = { id: 1, name: "rohit" };

console.log(obj.name);

console.log(obj["name"]);

// Modify The Member

obj.name = "rohit sharma"; // using dot operator

obj["id"] = 11; // using bracket operator

console.log(obj.name);

// Add More Members

obj.email = "rohit@gmail.com"; // using the dot operator

obj["mobile"] = "122121"; // using the bracket operator

console.log(obj);

// Delete the member

delete obj.id;

console.log(obj);

}

main();

// list json

function main() {

let list = ["delhi", "calcutta"];

// Modify the List :: Existing member

list[0] = "New Delhi";

list[1] = "Mumbai";

// Add New Item in the list

list.push("pune");

list.push("nagpur");

// Remove Item from the list

list.pop();

// Iterate all the element of list

for (let item of list) {

console.log(item);

}

}

main();

default function can be having alias name

XML HTTPREQUEST INTERVIEW POINT IMPORTANT

REACT always comes wtih default parameter e in member function

this e represent current object

default convetion of the parameter name is :: e

e -> very useful when working input element

to pass user defined parameter we have to call parameter

import { useState } from "react";

function App() {

return (

<div>

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

{/\* Event Demo is the name of the funciton , attributes are paramter\*/}

{/\* attribute and name of parameter can be anything \*/}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

<EventDemo imageId="237" title="first"/>

<hr />

<EventDemo imageId="238" title="second"/>

<hr />

<EventDemo imageId="239" title="third"/>

</div>

);

}

function EventDemo({imageId, title}) {

let [counter, setCounter] = useState(100);

let imageUrl = `https://picsum.photos/id/${imageId}/200/200`;

// console.log(imageId);

let likeMe = () => {

counter = counter + 1;

// Trigger the DOM Operation :: Asking the REACT to perform DOM

setCounter(counter);

};

// UI

return (

<div>

<img

src={imageUrl}

alt=""

style={{ width: "100%", objectFit: "cover" }}

/>

<h1>Like {counter}</h1>

<input type="button" value="Like Me" onClick={likeMe} />

</div>

);

}

export default App;

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

import { useState } from "react";

function App() {

return (

<div>

<ListDemo />

</div>

);

}

function ListDemo() {

// let list = ["237"]; // stateless

let [list, setList] = useState(["237"]); // stateful

let [imageInput, setImageInput] = useState("");

let addNewImage = () => {

let newList = [imageInput, ...list];

setList(newList);

};

let inputImageHandler = (e) => {

setImageInput(e.target.value);

}

return (

<div>

<h1>List Demo</h1>

<input type="text" placeholder="Enter Image Id"/ value={imageInput} onChange={inputImageHandler}>

<input type="button" value="Add New Image" onClick={addNewImage} />

{list.map((item, index) => (

<EventDemo key={index} imageId={item} title="first" />

))}

</div>

);

}

function EventDemo({ imageId, title }) {

let [counter, setCounter] = useState(100);

let imageUrl = `https://picsum.photos/id/${imageId}/200/200`;

let likeMe = () => {

counter = counter + 1;

// Trigger the DOM Operation :: Asking the REACT to perform DOM

setCounter(counter);

};

// UI

return (

<div>

<img

src={imageUrl}

alt=""

style={{ width: "100%", objectFit: "cover" }}

/>

<h1>Like {counter}</h1>

<input type="button" value="Like Me" onClick={likeMe} />

</div>

);

}

export default App;

to pass data from parent to child use of props and for vice versa we use callback

import {useState} from "react";

function App() {

return (

<div>

<EventDemo />

</div>

);

}

function EventDemo() {

// DATA MEMBER ::

// 1.STATELESS VARIABLE :: VALUE WILL CHANGE BUT UI WILL NOT UPDATE

// 2.STATEFUL VAIRABLE WHEN useState() is used

// 1. let background = "bg-primary";

\\ 2.

let [background,setBackground] = useState("bg-primary");

// MEMBER FUNCTION

let changeBackgroundPrimary = () => {

let newBackground = "bg-primary";

setBackground(newBackground);

background = "bg-primary";

alert();

}

let changeBackgroundSuccess = () => {

let newBackground = "bg-success";

setBackground(newBackground);

background = "bg-success";

alert();

}

let changeBackgroundDanger = () => {

let newBackground = "bg-danger";

setBackground(newBackground);

background = "bg-danger";

alert();

}

return (

<div>

<h1 className={`${background} text-light p-3`}>Event DEMO</h1>

<input

type="button"

className="btn btn-primary"

value="PRIMARY"

onclick={changeBackgroundPrimary}

/>

<input

type="button"

className="btn btn-success"

value="PRIMARY"

onclick={changeBackgroundPrimary}

/> <input

type="button"

className="btn btn-danger"

value="PRIMARY"

onclick={changeBackgroundPrimary}

/>

<p className="m-2 fs-4 ">

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Perspiciatis

tempora at debitis reprehenderit molestiae libero. Deserunt non

asperiores commodi saepe, fugiat est quae temporibus ex iusto

perferendis consequatur ab corporis eligendi vero vel alias eaque

aliquid id. Iste obcaecati, adipisci reiciendis nam non suscipit eos

repudiandae amet! Dolorum ratione maiores quas blanditiis deserunt

soluta ad, odit neque quo, est amet tempora officiis error corrupti

facere molestias similique explicabo velit quis cum veritatis, molestiae

unde illo? Odio amet doloribus labore aperiam qui voluptates at optio

quis repudiandae fuga officiis autem, iure, nesciunt dolorem nisi

nostrum est magnam alias aliquid, exercitationem ut?

</p>

</div>

);

}

export default App;

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

import {useState} from "react";

function App(){

return (

<div>

<EventDemo />

</div>

);

}

function EventDemo(){

// let title = useState("Hello World");

// console.log(title); // return array first param string ,2nd string is function

// declaring and destructing the stateful variable

let [title,setTitle] = useState("Hello World");

console.log(title); // return array first param string ,2nd string is function

return(

<div>

<h1>{title}</h1>

</div>

);

}

export default App;

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

import {useState} from "react";

function App(){

return (

<div>

<EventDemo />

</div>

);

}

function EventDemo(){

// let title = useState("Hello World");

// console.log(title); // return array first param string ,2nd string is function

// declaring and destructing the stateful variable

let [title,setTitle] = useState("Hello World");

// console.log(title); // return array first param string ,2nd string is function

// Member Function

let changeTitle = ()=>{

let newTitle = "Hi " + title;

// re-render : asking the REACT to perform DOM Operation

setTitle(newTitle);

};

return(

<div>

<h1>{title}</h1>

< input

type="button"

value="change the title"

onclick={changeTitle}

/>

</div>

);

}

export default App;

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

import {useState} from "react";

function App(){

return (

<div>

<EventDemo />

</div>

);

}

function EventDemo() {

// let counter = 100;

let [counter,setCounter] = useState(100);

// Member Function

let likeMe = ()=>{

counter += 1;

console.log(counter);

// Perform DOM Operation

setCounter

};

return(

<div>

<h1>Like</h1>

<h1>{counter}</h1>

< input

type="button"

value="Like Me"

onclick={likeMe}

/>

</div>

);

}

export default App;

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

import {useState} from "react";

function App(){

return (

<div>

<EventDemo />

</div>

);

}

function EventDemo() {

// let counter = 100;

let [counter,setCounter] = useState(100);

let [displayImage,setDisplayImage] = useState(https://picsum.phots/id/237/300/300");

// Member Function

let likeMe = ()=>{

counter += 1;

console.log(counter);

// Perform DOM Operation

setCounter

};

let changeImage = () =>{

displayImage = "https://picsum.phots/id/238/300/300";

// perform dom operation

setDisplayImage(displayImage);

}

let resetImage = () =>{

displayImage = "https://picsum.phots/id/237/300/300";

// perform dom operation

setDisplayImage(displayImage);

}

return(

<div>

<h1>Like {counter}</h1>

<img src="https://picsum.phots/id/237/300/300">;

<h1></h1>

< input

type="button"

value="Like Me"

onclick={likeMe}

/>

<br>

< input

type="button"

value="Change Image"

onclick={changeImage}

/>

< input

type="button"

value="Reset Image"

onclick={resetImage}

/>

</div>

);

}

export default App;

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

import {useState} from "react";

function App(){

return (

<div>

<EventDemo />

</div>

);

}

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

MAP OPERATOR

// <h1>Delhi</h1>

function EventDemo() {

let list = ["Mumbai","Pune","Nagpur","Tvm","Kochi"];

return(

<div>

<h1>List Demo</h1>

{/\* Any Valid Js Expression\*/}

{list.map((item)=>(

<h1>{item}<h1/>

))}

{list.map((item)=>(

<div className = "alert alert-primary mb-1">{item}</div>

))}

{list.map((item)=>item)}// simple string

</div>

);

}

export default App;

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

import {useState} from "react";

function App(){

return (

<div>

<EventDemo />

</div>

)

}

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

MAP OPERATOR

DYNAMIC LIST

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

// <h1>Delhi</h1>

function EventDemo() {

let [list,setList] = useState(["Mumbai","Pune","Nagpur","Tvm","Kochi"]);

let addNewItem = () => {

let newList = [ ...list , "Delhi" ];

// let newList = [ "Delhi" , ...list ];

setList(list);

}

return(

<div>

<h1>List Demo</h1>

<input type="button" value ="Add New Item" value="addNewItem"/>

<hr />

{list.map((item)=>(

<div className = "alert alert-primary mb-1">{item}</div>

))}

// here in above arrow funciton no curly braces because it is wrapped by jsx function

</div>

);

}

export default App;

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

import {useState} from "react";

function App(){

return (

<div>

<EventDemo />

</div>

)

}

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

MAP OPERATOR

DYNAMIC LIST through textbox

---->WE SHOULD NOT USE DOCUMENT OBJECT IN REACT <----

⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

// <h1>Delhi</h1>

function EventDemo() {

let [list,setList] = useState(["Mumbai","Pune","Nagpur","Tvm","Kochi"]);

let addNewItem = () => {

let newList = [ ...list , "Delhi" ];

// let newList = [ "Delhi" , ...list ];

setList(list);

}

return(

<div>

<h1>List Demo</h1>

<input type="text" placeholder="Add new city">

<input type="button" value ="Add New Item" value="addNewItem"/>

<hr />

{list.map((item)=>(

<div className = "alert alert-primary mb-1">{item}</div>

))}

// here in above arrow funciton no curly braces because it is wrapped by jsx function

</div>

)

}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Approach By dom

function EventDemo() {

let [list, setList] = useState(["todo1"]);

// Member Funcn

let addNewItem = () => {

// We should not use document object

let inputTag = document.querySelector("#inputid");

let inputValue = inputTag.value;

let newList = [inputValue, ...list];

setList(newList);

};

return (

<div>

<h1>List Demo </h1>

<input type="text" placeholder="Enter Todo" id="inputid" />

<input type="button" value="Add new Item" onClick={addNewItem} />

<hr />

{list.map((item) => (

<h1>{item}</h1>

))}

</div>

);

}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Approach 1

function EventDemo() {

let inputRef = useRef();

let [list,setList] = useState(["Mumbai","Pune","Nagpur","Tvm","Kochi"]);

let addNewItem = () => {

// let newList = [ ...list , "Delhi" ];

// let newList = [ "Delhi" , ...list ];

// setList(list);

console.log(inputRef.current);

let newValue = inputRef.current.value;

setList([newValue, ...list]);

// setList([inputRef.current.value, ...list]);

inputRef.current.value = "";

}

return(

<div>

<h1>List Demo</h1>

<input type="text" placeholder="Add new city" ref={inputRef} />

<input type="button" value ="Add New Item" value="addNewItem"/>

<hr />

{list.map((item)=>(

<div className = "alert alert-primary mb-1">{item}</div>

))}

// here in above arrow funciton no curly braces because it is wrapped by jsx function

</div>

);

}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Approach 2 -> add item in list eventhandler

function EventDemo() {

let [todo, setTodo] = useState("");

let [list, setList] = useState([]);

let inputHandler = (e) => {

console.log(e.target.value);

let newValue = e.target.value;

setTodo(newValue);

};

let addNewItem = () => {

let newList = [todo, ...list];

setList(newList);

setTodo("");

};

// UI

return (

<div>

<h1>List Demo </h1>

<input

type="text"

placeholder="Enter Todo"

value={todo}

onChange={inputHandler}

/>

<input type="button" value="Add New Item" onClick={addNewItem} />

<hr />

{list.map((item) => (

<h1>{item}</h1>

))}

</div>

);

}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Approach 3 -> add get list delete item event handler

function EventDemo() {

let [todo, setTodo] = useState("");

let [list, setList] = useState([]);

let inputHandler = (e) => {

console.log(e.target.value);

let newValue = e.target.value;

setTodo(newValue);

};

let addNewItem = () => {

let newList = [todo, ...list];

setList(newList);

setTodo("");

};

let deleteTodo = (index) => {

list.splice(index, 1);

console.log(list);

setList([...list]);

};

// UI

return (

<div>

<h1 className="bg-success text-white p-3">Todo App </h1>

<input

type="text"

placeholder="Enter Todo"

value={todo}

onChange={inputHandler}

/>

<input type="button" value="Add New Item" onClick={addNewItem} />

<hr />

{list.map((item, index) => (

<div key={index} className="d-flex justify-content-between">

<h1>{item}</h1>

<input type="button" value="DEL" onClick={() => deleteTodo(index)} />

</div>

))}

</div>

);

}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

export default App;

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

=============================

filename.mjs => modulejs file

=============================

(pls check from timestamp 9:40->22-05-2023)

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

basic fundamental of react no use of html

everything in react is JavaScript

every file is module in js

NOTE :->

File name => It starts with Capital Letter.

File name and function name are the same.

Make the module public.

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

App.js

function App(){

return "Hello World";

}

export default App;

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

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

App.js

function App(){

return "<h1>Hello World</h1>"; // THIS IS A STRING

}

export default App;

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

//⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️⬇️

JSX => JavaScript and XML X stands for XML

-- PREDEFINED TAG :: <h1> <p> <div> -> this are predefined

-- X :: Extended Markup Language

-- XML :: USER DEFINED TAG

-- USER DEFINED TAG : <App></App> or <App />

by using js function we created user defined tags

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

App.js

function App(){

return <h1>Hello World</h1>; //tag

}

export default App;

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

//returning more than one tag⬇️⬇️⬇️⬇️⬇️

>> ON SAVE

>> CODE FORMATTING

>> AUTO SUGGESTION

>> RUN THE PROGRAM :: npm start (there is no live server extension)

>> JSX :: ()

>> :: optional, Parethesis if html is one liner.

>> :: required, Parethesis for multiple line html.

>> :: required, Parethesis with two tags.

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

App.js

function App(){

return (<div><h1>Hello World</h1></div>); //tag

}

export default App;

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

>> Special Function :: Understood by REACT.

>> return JSX.

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

App.js

function App(){

return (

<div>

<h1>Hello World</h1>

<hr/>

<Header>

<hr/>

</Header>

</div>);

}

/\*

Simple Js function

function simpleFunc(){

}

\*/

// USER DEFINE TAG

// <Header />

// Simple JSX function

>> this funcion is reusable

function Header(){

return(

<div>

<h1>This is header</h1>

</div>)

}

function Footer(){

return(

<div>

<h1>This is footer</h1>

</div>)

}

function MainBody(){

return(

<div>

<h1>

Lorem Lorem Lorem Lorem Lorem Lorem Lorem Lorem Lorem Lorem

Lorem Lorem Lorem Lorem Lorem Lorem Lorem Lorem Lorem Lorem r</h1>

</div>)

}

export default App;

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

`````````````````````````````````````````````

class is keyword in javascript

className is property for purpose of styling

`````````````````````````````````````````````

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

interpolation basics

function InterpolationBasic(){

//1. Data member

let title="fard";

let para ="fruygaedyufaerdjcvuyefrdghshuyfreuyfyf

ruydfgkugfruuyfgrefruyefruerruef

hiklruhiuarerfhrue"

let price = 99.99;

let user = {id:1.name:"abc",email:"abc@gmail.com"};

//2. member function

//3. UI/view

return(

<div>

<h1>{title}</h1>

<p>{para}</p>

<h1>{price}</h1>

<h1>{user.id}--{user.name}--{user.email}</h1>

<h1>

{

/\*ANY VALID JS SYNTAX\*/

JSON.stringify(user)

}</h1>

</div>

)

}

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

interpolation basics 2

function InterpolationBasic(){

//1. Data member

let id = "id1";

let mytheme = "bg-success text-light p-3";

let title="fard";

let para ="fruygaedyufaerdjcvuyefrdghshuyfreuyfyf

ruydfgkugfruuyfgrefruyefruerruef

hiklruhiuarerfhrue"

let price = 99.99;

let user = {id:1.name:"abc",email:"abc@gmail.com"};

//2. member function

//3. UI/view

return(

<div>

<h1 className={mytheme} id="id1">{title}</h1>

<p>{para}</p>

<h1 className={mytheme} id={id}>{price}</h1>

<h1>{user.id}--{user.name}--{user.email}</h1>

<h1>

{

/\*ANY VALID JS SYNTAX\*/

JSON.stringify(user)

}</h1>

<p className={mytheme}>{para}</p>

<p className={mytheme}>{para}</p>

<p className={mytheme}>{para}</p>

<p className={mytheme}>{para}</p>

<p className={mytheme}>{para}</p>

</div>

)

}

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

interpolation basics 3

function InterpolationBasic(){

//1. Data member

let id = "id1";

let mytheme = "bg-success text-light p-3";

let title="fard";

let para ="fruygaedyufaerdjcvuyefrdghshuyfreuyfyf

ruydfgkugfruuyfgrefruyefruerruef

hiklruhiuarerfhrue"

let price = 99.99;

let user = {id:1.name:"abc",email:"abc@gmail.com"};

let mystle={color:"red"};

//2. member function

//3. UI/view

return(

<div>

<h1 className={mytheme} id="id1">{title}</h1>

<p>{para}</p>

<h1 className={mytheme} id={id}>{price}</h1>

<h1>{user.id}--{user.name}--{user.email}</h1>

<h1>

{

/\*ANY VALID JS SYNTAX\*/

JSON.stringify(user)

}</h1>

<p className={mytheme}>{para}</p>

<p className={mytheme}>{para}</p>

<p style = {{color:"red"}}>{para}</p>

<p style = {{color:"red",padding:"16px",background:"tomato"}}>{para}</p>

<p style = {mystyle}>{para}</p>

</div>

)

}

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

running js in console

node fname.js

console.log(argument);

console.log(parameter);

console.log(p1,p2,p3);

if(0) -> false

1 Setup VS Code

2 Create an account on Git hub.

3 Setup Postman API

Jayaram M S to Everyone 15:59

Setup Node JS

Make different types of HTPP call using postman, GET ,PUT,POST DELETE.

Write an HTML program to Print your name using VS Code.

node -v

npm -v

npx -v

github pass Tejas@hub3

to put gap between two contents put margin

inline css properties get priority

div class='' style

by use of flex contents are adjacent to each other no blocking

resume and hello sum helllo world