27. Create an 8 \* 8 chessboard UI using html, css and js. Each box size will be 20px\*20px. Write modular code so that it is easier to extend. Modular means reusable and less of repetition

26. Write a recursive JS function which takes a JSON object and prints all the values present in the object.

Input = { a: 1, b: 2 }

Output = 1 2

Input = { b: { c: 2} }

Output = 2

// your code here which prints all the values of the JSON object

const printJson = (jsonObj) => {}

Note: Don't use simply JSON.stringify(jsonObj) as it would just return objects in stringified notation. We only want values in JSON objects

**Solution**: function ab(obj) {

        for (let key in obj) {

            if (typeof obj[key] === 'object') {

                ab(obj[key]);

            } else {

                console.log(obj[key]);

            }

        }

    }

    let input = { a: 1, b: 2 };

    let input2 ={b: {c:2}};

    ab(input);

    ab(input2);

node json.js run terminal

write the code to find 2 max from the given array.

let arr = [1,4,2,3,5];

let n = arr.length;

arr.sort();

let result = arr[n-2];

console.log(result)

Q. Fetch the data from the API?

Code for converting string into an array

                    i/p: "i like java"

         o/p: ["i", "like", "java"]

let string = "i like java";

let arr = string.split(" ");

console.log(arr)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Document</title>

</head>

<body>

    <h1 id="h1">0</h1>

    <button id="incbtn">Increase</button>

    <button id="decbtn">Decrease</button>

    <button id="resetbtn">Reset</button>

    <script src = "counter.js"></script>

</body>

</html>

let incbtn = document.getElementById("incbtn");

let decbtn = document.getElementById("decbtn");

let resetbtn = document.getElementById("resetbtn");

let h1 = document.getElementById("h1");

incbtn.addEventListener("click" , ()=>{

     h1.innerText++

})

decbtn.addEventListener("click" , ()=>{

    h1.innerText--

})

resetbtn.addEventListener("click" , ()=>{

    h1.innerText = 0;

})

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Document</title>

</head>

<body>

    <input type="text" id="fname" placeholder="First name">

    <input type="text" id="lname" placeholder="Last name">

    <input type="email" id="email" placeholder="Email">

    <button id="btn">ADD</button>

    <script src ="formns.js"></script>

</body>

</html>

let fname = document.getElementById("fname");

let lname = document.getElementById("lname");

let email = document.getElementById("email");

let btn = document.getElementById("btn");

btn.addEventListener("click" , fx);

function fx(){

    // alert(`${fname.value} ${lname.value} ${email.value}`)

    alert(fname.value + " " + lname.value + " " + email.value);

}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Document</title>

    <style>

        table, td, th {

          border: 1px solid black;

        }

        table {

          border-collapse: collapse;

          width: 50%;

        }

        </style>

</head>

<body>

    <table id="container">

        <tr>

            <td>ID</td>

            <td>NAME</td>

            <td>GENDER</td>

            <td>EMAIL</td>

            <td>STATUS</td>

        </tr>

    </table>

    <script src= "./fetch.js"></script>

    <!-- <script src= "./script.js"></script> -->

</body>

</html>

let table = document.getElementById("container");

async function fetching(){

    let response = await fetch("https://gorest.co.in/public/v2/users");

    let data = await response.json();

    console.log(data);

    data.map((item) => {

        let row = document.createElement("tr");

        row.innerHTML = `<td>${item.id}</td> <td>${item.name} </td> <td>${item.gender}</td> <td>${item.email}</td> <td>${item.status}</td>`

        table.append(row);

    })

}

fetching()

<!-- //sort by marks -->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Document</title>

</head>

<body>

    <script src ="marks.js"></script>

</body>

</html>

//sort by marks

const users = [

    {id: 1, name: 'A', mark: 80},

    {id: 2, name: 'B', mark: 40},

    {id: 3, name: 'C', mark: 90},

    {id: 4, name: 'D', mark: 30},

    {id: 5, name: 'E', mark: 70},

    {id: 6, name: 'F', mark: 50},

    {id: 7, name: 'G', mark: 60},

    ];

    users.sort((a,b)=> {return (a.mark-b.mark)})

    users.sort((a,b)=> {return (b.mark-a.mark)})

    console.log(users);

//modify by marks pass fail

const users = [

    {id: 1, name: 'A', mark: 80},

    {id: 2, name: 'B', mark: 40},

    {id: 3, name: 'C', mark: 90},

    {id: 4, name: 'D', mark: 30},

    {id: 5, name: 'E', mark: 70},

    {id: 6, name: 'F', mark: 50},

    {id: 7, name: 'G', mark: 60},

    ];

    users.map(adding)

function adding(){

for(let i=0;i<users.length;i++){

if(users[i].mark>=60){

users[i].Result="Pass"

}

else{

users[i].Result="Fail"

}

}

}

console.log(users);

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Document</title>

</head>

<body>

    <script src ="passmarks.js"></script>

</body>

</html>