JAVASCRIPT

// Reffer in MDN website

// trim() this function used to trim the white space

let name= "sekar ";

console.log(name.trim().length);

// Generate randon text in my name "SEKAR"

const i=Math.floor(Math.random()\*4 +1);

console.log("sekar".charAt(i));

// Effective code

const myName="sekar"

console.log(myName.charAt((Math.floor(Math.random()\*myName.length))));

// ternary operator find grade

let mark=90;

mark>=90 ? console.log("A") : mark>=80 ? console.log('B') : console.log('F');

// CHAPTER 10 : USER INPUT

//ALERT

let at = alert("hello")

console.log(at);

// CONFIRM

let con= confirm('are u satisfy');

console.log(con);

//PROMPT

let pro = prompt('enter name')

console.log(pro ?? "you did not enter ur nam");// ?? null colasking operrator used to remove null value

let pro1 = prompt('enter name')

if(pro1){

console.log(pro1.trim() ?? "you did not enter ur name")// trim function is used to remove the white spaces in our name

}

else{

console.log("you did not enter ur name");s

}

// VAR LET CONST DIFFERENT

/\*var is functon scope

let & const is block scope .It work only wthin the block\*/

//ARRAY

{

const myArray =[];

myArray[0]="sekar";

myArray[1]="pvs";

myArray[2]=100;

myArray[3]=true;

console.log(myArray);

// ARRAY METHOD PUSH,POP,UNSHIFT,SHIFT

//PUSH is used add new element in last

myArray.push("newOne push");

console.log(`PUSH ${myArray}`)

//POP is used delete element on last

myArray.pop();

console.log(myArray);

//UNSHIFT is used to add new element in first

myArray.unshift("newOne unshift")

console.log(myArray);

// SHIFT is used to delete element on first position

myArray.shift();

console.log(myArray);

// SPLICE METHOD is used to delete and replace the element

myArray.splice(2,1)//delete the second element in Array

console.log(myArray);

// using SPLICE we replace the element

//ex1

myArray.splice(2,1,200)//replace the second element in Array

console.log(myArray);

//ex2

myArray.splice(2,1,"splice replace")//replace the second element in Array

console.log(myArray);

//join ()method is used to join

const Join = myArray.join();

console.log(Join);

// split used to split

const Split = Join.split(",");

console.log(Split);

// joining or adding two array using concat() method || using SPREED(...) operator to adding or joining arrar

const myArray1 = ["hi","hello", 7];

console.log(myArray1);

const newArray = myArray.concat(myArray1)

console.log(newArray);

// SPREED OPERATOR(...)

const spreedArray =[...myArray,...myArray1]

}

// OBJECT IN JAVA SCRIPT

const myObj ={

name :"sekar",

age:22,

subject:{

sub1:"html",

sub2:"css",

sub3:"js"

},

class:"mca",

action: ()=>{

return('hello world');

},

action1: function(){

return `using value same object use this keyword ${this.name} `

}

};

console.log(myObj.name);

console.log(myObj.age);

console.log(myObj.subject);

console.log(myObj.subject.sub1);

console.log(myObj.subject.sub2);

console.log(myObj.action());

// INHERITANCE OF OBJECT

const car= Object.create(myObj);

console.log(car.action());

//for in loop

const movie ={

actor :"sekar",

actress:"?",

music:"aniruth",

director:"lokesh"

}

for(job in movie){

console.log(movie[job]);

}

// if we want delete the key and key value using delete

delete movie.actress;

// DESTRUCTURE THE OBJECT

const {actor:myFavActor} = movie;

// OR

const {actor,music,director}=movie;

//CLASSES

//simple object

const myPizaa={

size : "medium",

crust: "original",

bake : function (){

return console.log(`the pizza in ${this.size} size and in ${this.crust}`);

}

}

myPizaa.bake();

// conver into a class

// simple

class pizza{

constructor(){

this. size="medium";

this.crust ="orignal";

}

bake=()=>{

return console.log(`the pizza in ${this.size} size and in ${this.crust}`);

}

}

const Mypizza= new pizza();

console.log(Mypizza.crust);

console.log(Mypizza.bake());

// same code but intrective manner

class pizza1{

#sause = "chill" // private property npt not this fixed value

constructor(sizePizza,crustpizza){

this. size=sizePizza;

this.crust =crustpizza;

}

bake=()=>{

return console.log(`the pizza in ${this.size} size and in ${this.crust}`);

}

}

const Mypizza1= new pizza1("medium","orignal");

console.log(Mypizza.bake());

// child class

/\* syntax

class childclass\_Name extends parent class\_name(){

constructor(){

}

}

\*/

class childPizza extends pizza1{

constructor(sizePizza,crustpizza){

super(sizePizza,crustpizza);

}

}

const childObj = new childPizza();

console.log(bake());

// JSON IN JAVASCRIPT

const jsonObj = {

name : "sekar",

sub :['HTML','CSS','JS'],

function:()=>{

console.log("Thank you for subscribe ");

}

}

// OBJECT CONVER INTO JSON

const sendJson = JSON.stringify(jsonObj);

console.log(sendJson);

// JSON CONVER INTO OBJECT

const receiveJson = JSON.parse(sendJson);

console.log(receiveJson);

// ERROR HANDLING

/\* Type of error

1. Refference error

2. Syntax error

3. Type error

\*/

const makeError =()=>{

try{

const name = "sekar";

name = "pvs";

}catch(err){

console.error(err);

console.table(err); // Through the error in table form

console.warn(err);

}

};

makeError();