## JSON

* JSON stands for JavaScript object notation
* JSON is a simple format to store data
* In object key value pair place between quotes or Single quotes but better is that you use single quotes and write number in value without quotes and in array every except number should put in quotes
* JSON objects and Arrays place between SINGLE QUOTES ONLY
* DOUBLE QUOTES INSIDE AND SINGLE OUTSIDE

//JSON objects must be

const obj = '{"name":"adil","father":"younas","class":"MSC"}'

//JSON Array must be

const array = '["adil","aqil","inter"]'

## JSON to Object JSON.parse()

* Single Quotes must require

//JSON objects must be

const obj = '{"name":"adil","father":"younas","class":"MSC"}'

let data1 = JSON.parse(obj)

console.log(data1);

//JSON Array must be

const array = '["adil","aqil","inter"]'

let data = JSON.parse(array)

console.log(data);

# Object to JSON JSON.stringify()

//JSON objects must be

const obj = {name:"adil",father:"younas",class:"MSC"}

let data1 = JSON.stringify(obj)

console.log(data1);

//JSON Array must be

const array = ["adil","aqil","inter",20]

let data = JSON.stringify(array)

console.log(data);

## Rest Operator or rest parameter (receiving via parameter)

Important Note: ☠️

let arrr = [1,2,3]

console.log(...arrr); //array value

console.log(arrr); // array

* The rest parameter **allows us to represent an indefinite number of arguments as an array**. By using the rest parameter, a function can be called with any number of arguments.
* function will only work when count of parameters = count of arguments

function sum(a,b,c){

  console.log(a+b+c);

}

sum(2,1,5) //8

sum(2,3,1,4) //6 instead of 10

* If you want to add arguments and saw the result then you should try this method
* The Word Arguments Here Is Necessary
* This Method Is Limited With Numbers Only

function sum(){

  let sum = 0

  for(let key in arguments){

    sum += arguments[key]

  }

  console.log(sum);

}

sum(1,2,3,4,5,6) //21

sum(1,1,1) //3

//real use of rest operator

function sum(name,classs,...numbers){

  let sum = 0

  for(let key in numbers){

    sum += numbers[key]

  }

  console.log(`name is ${name} class is ${classs}

   and total marks are ${sum}`);

}

sum("adil younas", "MSC", 1,2,3,4,5,6)

//name is adil younas class is MSC and total marks are 21

sum("younas", "php", 1,2,3,4,5,6)

//name is younas class is php and total marks are 21

Rest operator with object destructuring

const students = {

  name:"ajay",

  age:25,

  hobbies:["cricket","Singing"]

}

const {age,...rest} = students

console.log(age,rest);

const newStudent = {

  ...students,

  age:12

}

console.log(newStudent);

Warning:

if you are trying to use …numbers in middle of in start then it will return an error

Important note:

function sum(...arg){

  console.log(arg); //array

  console.log(...arg); //values of array

}

sum(1,2,3,1,3,1)

Another example of rest operator

function fun(a,b,...c){

console.log(`${a} , ${b}`);

console.log(c);

console.log(c[0]);

console.log(c.length);

console.log(c.indexOf("Edan"));

}

fun("Ronaldo","Neymar","Pele","Messi","Edam")

// Ronaldo , Neymar

//  (3) ['Pele', 'Messi', 'Edam']

//  Pele

//  3

// -1

Very important point for iteration ☠️

Arr[key] = value of array

for in loop iteration

let arr = [10,2,1]

function fun(arr){

let sum = 0

for(let key in arr){

  sum+= arr[key]

}

console.log(sum);

}

fun(arr)//13

for of loop iteration

key = value of array

let arr = [10,2,1]

function fun(arr){

let sum = 0

for(let key of arr){

  sum+= key

}

console.log(sum);

}

fun(arr)

# Spread Operator (pass object or array via argument of function or giving via argument)

* When you give triple dots in parameters then it is rest operator or rest parameter
* When you give triple dots in argument then it is called spread operator
* Spread operator replace the call, apply, bind function

What is the call, apply, bind function

const students = {

  name:"ajay",

  age:25,

  hobbies:["cricket","Singing"]

}

const teachers = {

  name:"nasir",

  age:50,

  hobbies:["beat children","unnecessary laugh"]

}

function fun(a,b,c){

console.log(`${this.name} and ${this.hobbies} ${a} ${b} ${c}`);

}

fun.call(teachers,"react")

const arrr = [1,3,1]

fun.apply(students,arrr)

So now what is Spread operator

function fun(a,b,c){

  console.log(` value : ${a}   value :  ${b}   value :  ${c}`);

}

let arrr = [1,2,3]

fun(...arrr)

//Three will cause an error

//so don't try this at home be safe

fun.call(...arrr)

fun.apply(...arrr)

fun.bind(...arrr)

Important Note: ☠️

let arrr = [1,2,3]

console.log(...arrr); //array value

console.log(arrr); // array

# Destructuring

Note1: Instead of storing data in single variable we use destructuring

Note2: for taste you can use rest operator too

Note3: student.name = student[“name”] in objects

//Create Object and Array and then store it in variable

const students = {

  name:"ajay",

  age:25,

  hobbies:["cricket","Singing"],

  father:"younas"

}

let a = students.name

let a1 = students["name"]

console.log(a);

console.log(a);

const fruits = ["Apple","banana","Avocado"]

let b = fruits[0]

console.log(b);

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

//if you want to store data in variable then destructuring may help you

//object destructuring

const students2 = {

  name:"ajay",

  age:25,

  hobbies:["cricket","Singing"],

  father:"younas",

  start:function(){

    console.log(`Start the log programe`);

  },

  class:"MSC",

  city:"Gujranwala"

}

let {name,age,hobbies,father,start,...rest} = students2

console.log(start,age);

console.log(rest);

//Array Destructuring

const DryFruits = ["Avocado","HP","Gilgit Baltistan"]

let [v1,v2,...v3] = DryFruits

console.log(v1,v3);