**Ecma script (Advanced javascript study)**

**Mutable and immutable**

**mutable objects and arrays**

**Immutable string and integer…**

**Spread operator**

**Used to allocate copy of an object created so that it will not alter its**

**Orginal values…**

const globalPlayers=

{

id:23,

name:"ronaldo",

powerLevel:100

};

const player1={...globalPlayers};// global player is assigning copy

player1.powerLevel=50;

console.log(globalPlayers);

spread operator {…} three dots in front of variable it will give copy of

globalPlayers to player1

const cricketPlayers=["Anil","govasker","dhoni"];

const cricketPlayers2=["kumbele","tendulkar","vasim"];

let playerList=cricketPlayers.concat(cricketPlayers2);

concatenating cricketPlayers to cricketPlayers2

This can be done by spread operators

let spreadList=[…cricketPlayers,…cricketPlayers2];

pushing value to array

spreadList=[…cricketPlayers,”Arun”];

spredList=[...cricketPlayers,"Arun"];//pushing data to array using spred

spredList=[...cricketPlayers,cricketPlayers2];//concatinating an array

console.log(spredList);

**Rest**

In an array if we are printing its zeros position value then we need to give a comment like array[0] instead of this we can use (array destructuring)

const globalPlayer=['Rahul','ajay','sunil']

console.log(globalPlayer[0]);

const [first,second,third]=globalPlayer;//array destructuring

console.log(first);

console.log(first); this will give value in zeroth position of array…

spread operator gives a copy of existing variable thus any changes will not affect parent values…..

Rest will also give copy of a variable from an object

As shown below:

const playerDtls={

    id:1,

    name:"Arun",

    type:"cricket"

};

const{id, ...rest}=playerDtls;

console.log(id);

console.log(rest);

here we are only using id thus we are keeping our other variables as

Rest(which is kept as reference for any future use).Other variables like name and type in object playerDtls are kept as a copy in rest.

Arrow

const myFunc=()=>console.log("Running")//function can be written like this

myFunc()

const sum=(value)=> { return value+5}//function passing value

console.log(sum(10))

foreach method

const listPlayers=[{

    id:1,

    name:"Aswin",

    age:23

},

{

    id:2,

    name:"Theertha",

    age:22

}

,{

    id:3,

    name:"alaka",

    age:24

}]

// for(let i=0;i<listPlayers.length;i++)

// {

//     console.log(listPlayers[i].name);

// }

//foreach ,map,filter refer

listPlayers.forEach((value,index,array)=>

{console.log(value)})

listPlayers.forEach((value,index,array)=>{console.log(value)})

for each is used to iterate values in a variables

refer this and study later [Javascript Advance 2020 malayalam tutorials | ES2020 | ജാവാസ്ക്രിപ്റ്റ് വെബ് പ്രോഗ്രാമിംഗ് - YouTube](https://www.youtube.com/watch?v=TxFXiPe8Wt8&list=PL4KwFGqvN4nsQMUNcSLFzPhf3P8qegRqX&index=5)