**Datatypes:**

* Primitive datatype(Eg:number,String)
* Non-primitive datatype(Array,Object)

**Primitive datatype(Call by Value)**

//value-(primitive-oru ethulaa update anaa ella idthalayum agathu)

Eg:

let a=10;

let b=a;

a=20;

console.log(b);

**Array Functions:**

1. map()-applies mathematical logic to each element of an array and returns modified array(eg:calculate multipleof2,multipleof3)
2. filter()-condition checking.(eg:to find odd,even,multipleof 5 etc).
3. reduce()-operation perform panni ,single valuaa return pannum(eg:sum,multiply etc).

**1.map()=function passed as parameter(callback)**

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

//applies functions and returns as array

let doubledarray=arr.map((n)=>2\*n);

console.log(arr);

console.log(doubledarray);

**Output:**

[ 1, 2, 3, 4, 5 ]

[ 2, 4, 6, 8, 10 ]

**2.filter()**

**Eg:1 – Odd Elements**

let arr1=[1,2,3,4,5];

//odd will be added ,even will not be added,and it returns odd array

let odd=arr.filter((n)=>{return n%2!=0});

console.log(arr);

console.log(odd);

**Output:**

[ 1, 2, 3, 4, 5 ]

[ 1, 3, 5 ]

**Eg:2-Multiples of 5**

let arr3=[13,15,20,35,27];

let multiplesof5=arr3.filter((n)=>{return n%5==0;});

console.log(arr3);

console.log(multiplesof5);

**Output:**

[ 13, 15, 20, 35, 27 ]

[ 15, 20, 35 ]

**3.reduce()**

**Eg:1**

//1st parameter is array iterating element

//2nd parameter is sum value stored place while iterating

//the one present after closed set bracket is initial value of sum

let arr4=[1,2,3,4,5,6];

let sumofarray=arr4.reduce((n,sum)=>{

return n\*sum;},1);

console.log(sumofarray);

**Output:**

720

**Eg:2**

//arr5-stores actual arr value

let arr5=[1,2,3,4,5];

//arr6-stores the array value of arr5,where each element is multiplied by 3

let arr6=arr5.map((n)=>n\*3);

//arr7-stores the number which is even

let arr7=arr6.filter((n)=>{return n%2==0});

//sumofarray1-stores sum of all elements

let sumofarray1=arr7.reduce((sum1,n)=>{return sum1+n;},0);

console.log(arr5);

console.log(arr6);

console.log(arr7);

console.log(`Sum:${sumofarray1}`);

/\*

let arr6=arr5.map((n)=>n\*3)

             .filter((n)=>n%2==0)

             .reduce((sum,n)=>{return sum\*n;},0);

console.log(arr6);

\*/  
  
**Output:**

[ 1, 2, 3, 4, 5 ]

[ 3, 6, 9, 12, 15 ]

[ 6, 12 ]

Sum:18

**Destructuring-assigning array elements to variables in single line**

**Eg:1**

//Destructuring-assigning variables to some other variables in one line

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

const [one,two,three,four,five]=b;

console.log(three);

**Eg:2 – Rest Operator**

//In first variable 1 number will be stored,and in second variable rest of the elements are assigned.

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

const [one,...two]=b;

console.log(two);

**Eg:3-Spread Operator**

//adding extra elements to an array using spread operator

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

let num=[0,...b,6,7];

console.log(num);  
  
**Eg:4-Spread Operator-merging arrays and objects**

**Eg:**

const obj1={x:3,y:4};

const obj2={y:5,z:6};

//while merging one object will be added first and then second object will be addec ,so the second object

const spread={...obj1,...obj2};

console.log(spread);