Array :-

```
// let marks1 = 100;
// let marks 2 = 50;
// let marks 3 = 70;
// let marks 4 = 80;
```

or

```
let marks = [100, 50, 70, 80, 90];

console.log ( marks ); Isme saare element print ho ga.
console.log( marks.length ); Isme array ka length print hoga.
```

Push :-

```
let arr = [100, 50, 80 ];
arr.push ( 90 );
arr.push ("strike");

console.log(arr);

// pop : delete element from end.

arr.pop ();
console.log (arr);
```

* Unshift :- Starting me add karna hai.

~~arr.unshift()~~

```
arr.unshift(10);
console.log(arr);
```

* Shift :- Starting me delete karna hai.

```
arr.shift();
console.log(arr);
```

Loops:

```
let arr = [10,30,50,90,11];

// for (let i=0; i<arr.length; i++) {
//     console.log(arr[i]);
// }
```

or

```
for (let num of arr) {
    console.log(num);
}
```

Isko bolte hai for of loop.

```
let arr = [10, 30, 50, 90, 11];
let arr2 = arr;

arr2.push(70);

console.log(arr);
```

object reference ko copy karta hai.

* Slice.

```
const arr = [10, 30, 50, 90, 11];
const arr2 = arr.slice(2, 4);
console.log(arr);
```

* Splice : original arr me se change hoga.

```
const arr = [10, 30, 50, 90, 11];
arr.splice(1, 3);
console.log(arr);
```

Isme se 1st index se lekar 3rd index matlb 30, 50, 90 to remove kar do to bchega [10, 11].

Note: splice me hm original array me changes karte hai. par slice me original array me changes nhi karte.

Concat.

```
const arr = [10, 30, 50, 90, 11];
const arr2 = ["Rohit", 11, true];
const arr4 = [90, 1, false];

const arr3 = arr.concat(arr2, arr4);

console.log(arr3);
```

Spread operator:-

```
const arr3 = [...arr, ...arr2, ...arr4];
console.log(arr3);
```

↳ Three array ko saath one kata hai.

Join.

```
const names = ["Alice", "Bob", "Charlie"];
console.log(names.join("-"));
```

(jiske jagah pe '-' bhi dal skte hai.)

output:- Alice-Bob-Charlie.

1) ~~Simple searching~~ :-

2) Sorting :-

```
const names = ["Alice", "Rohit", "Bob", "Mohit", "charlie"]

names.sort();
console.log(names);
```

output :- 'Alice', 'Bob', 'charlie', 'Mohit', 'Rohit'

3) reverse :

```
names.reverse();
console.log(names);
```

output :- ['charlie', 'Mohit', 'Bob', 'Rohit', 'Alice'].

* descending order :- pehle sort karo phir reverse kar do.

Eg :-
```
const names = ["Alice", "Rohit", "Bob", "Mohit", "charlie"];

names.sort();
names.reverse();
console.log(names);
```

output :-

['Rohit', 'Mohit', 'charlie', 'Bob', 'Alice']

In 3-D Array.

```
Const arr = [10,30,50,[70,90,[60,19,99],11],20];
Console.log(arr[3][2][1]);
```

arr[3] → phele array ke liye
arr[2] → 2nd array ke liye
arr[1] → ye 3rd array ke liye.

Another method.

* flat.

```
Const a = arr.flat(Infinity);
Console.log(a);
```

Infinity → ye level based hota. matlb kitne level tak array ko flat karna hai. 1, 2, 3, ... ∞