

A | 0 | < | 3 | 0 |

Chapter 4 In JS

⇒ Arrays

collection of items:- ⇒ linear method
to store information.

let movies = ["ironman", "hulk", "thor"]

let marks = [97, 82, 56, 36, 82]

difference b/w Property & method

Property is like which we
store and do nothing.

Other side method is the
function which performs
property.

12

type of arrays:-

type of array is object

~~not~~

Array in Java \Rightarrow
 \Rightarrow mutable

strings \Rightarrow immutable

let theories = [97, 98, 14, 13, 15]

```
for (let index = 0; index < theories.length; index++)  
{  
  console.log(theories[index]);  
}
```

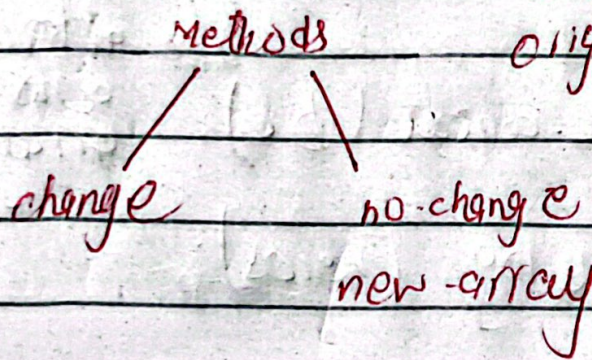

Array Methods:-

Add at the last of array.

Push (): add to end \Rightarrow change in original array

Pop (): delete from end & return
 \downarrow change in original array

toString (): converts array to string
 \Downarrow do not change in original array



```
let marks = [ . . . . . ]
```

```
marks.push("97")
```

```
marks.pop ( );
```

```
let delete = marks.pop ( );
```

```
console.log(marks)
```

```
console.log(marks.toString());
```


concat() :- donot change in original array.
join multiple arrays & returns result

```
let let dc = [thor, ironman, Ahsoka]
```

```
let bc = [sلمان, Talha]
```

```
let heroes new = dc.concat(bc); also attach another array.
```

```
console.log (heroes);
```

unshift :- Add to start
look like push

```
let dc = [thor, batman]
```

```
dc.unshift("antman");
```

```
console.log ( );
```


shift () :- ~~not~~ start
delete" and
return the result and
also change in original array.

```
let dc = shift (1);  
console.log (1);
```

→ slice () :- ^{don't change in original array.} returns a
piece of the
array, don't change
in original array.
slice (startIdx, endIdx)

```
let dc = [ 'Thor', 'Spiderman', 'Batman' ]
```

```
console.log (dc.slice (2, 4));
```

console

(1) ^{not include.} → start - end.
slice () →

splice(): change original
array

(add, remove, replace)

splice(start index, delcount, new value);

[1, 2, ~~X~~, ~~X~~, 5, 6, 8]
0 1 2 3 4 5 6

~~splice~~
splice(2, 0, 101, 102)

[1, 2, 101, 102, 5, 6, 8]
0 1 2 3 4 5 6