



Date / /

Arrays:-

Arrays in JavaScript:-

Collection of items → Linear way to arrange items.

↳ similar type items.

creation of array

Let Array-Name = [];

Let Marks = [1, 2, 3, 4, 5];

Let Heroes = ["ironman", "hulk", "thor"];

To find the length of the array we can use the property - Arr-Name.length.

Array is a type of object which have key-value pairs but instead of key it's ~~index~~ index. there is index.

Array indices:

Suppose, we have an array named marks.

Let marks = [22, 23, 24, 25, 26];

0 1 2 3 4

To access ~~the~~ a particular item in the array we need to know the index and then we can access that item through Arr-Name [index];



To change the value of the particular indices of the array then we can do is:-

Arr[Name [Index value]] = change value.
Marks [5] = 77;

Arrays are mutable.

Array methods:-
~~~~~

1) Push(): Adds to end;

```
let Veggies = ["Brinjal", "Tomato"];  
Veggies.push("carrot");
```

↳ carrot will be added to the end of Veggies array.

2) Pop(): Deletes from end;

```
Veggies = ["Brinjal", "Tomato", "carrot"]  
Veggies.pop();
```

↳ deletes the element from end.

3) toString(): Converts the array to string  
Veggies.toString();

↳ converts the arrays to string.

4) Concat(): Joins multiple arrays & returns result.

```
let marks = [1, 2, 3, 4];  
Veggies.concat(marks);
```

↳ concatenates marks array with Veggies.





5) `unshift()`: Adds to start equivalent to push methods only difference is it adds to start.

`marks.unshift(44);`

`marks = [44, 1, 2, 3, 4];`

6) `shift()`: deletes element from first similar to pop.

~~`marks.pop()`~~  
`marks.shift();`

`marks = [1, 2, 3, 4];`  
          0 1 2 3

7) `slice` method

Returns a piece of the array.

`slice (start index, end index);`

`let slicedArray = marks.slice(1, 3);`

`marks = [2, 3]`

The end index is non inclusive.