

Declaration:

```
let a = [1,2,3,4,5];  
console.log(a.length);
```

Output: 5

Accessing Array Elements:

```
console.log(a[2]);
```

Output: 3

Note that typeof an array is an object.

Example:

```
let a = [1,2,3,4,5];  
console.log(typeof(a));
```

Output: object

Changing value of arrays:

Strings are immutable, arrays are mutable. In case of arrays we can change the value of arrays, but in case of strings it is not possible.

Example 1:

```
let a = [1,2,3,4,5];
```

```
a[0] = 100;
```

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

Output: [100, 2, 3, 4, 5]

Example 2:

```
let a = "Abhra";
```

```
a[0] = 100;
```

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

Output: Abhra

This was possible because arrays are mutable but strings are immutable.

Array Methods:

1. toString():

Converts an array to a string and returns the result. The elements of the array are converted to strings and separated by commas.

Example:

```
let fruits = ["Apple", "Banana", "Mango"];
let fruitsString = fruits.toString();
console.log(fruitsString);
```

Output: Apple,Banana,Mango

2. join():

Creates and returns a new string by concatenating all the elements in an array, separated by a specified separator string. If no separator is provided, a comma (,) is used by default.

Example 1:

```
let fruits = ["Apple", "Banana", "Mango"];
let fruitsString = fruits.join();
console.log(fruitsString);
```

Output: Apple,Banana,Mango

Example 2:

```
let fruits = ["Apple", "Banana", "Mango"];
let fruitsString = fruits.join(' and ');
console.log(fruitsString);
```

Output: Apple and Banana and Mango

3. pop():

Removes the last element from an array. "This method changes the length of the array".

Example 1:

```
let fruits = ["Apple", "Banana", "Mango"];
let lastFruit = fruits.pop(); // stores the popped element
console.log(lastFruit);
console.log(fruits);
```

Output:

```
Mango  
[ 'Apple', 'Banana' ]
```

4. push():

Adds one or more elements to the end of an array. This method modifies the original array.

Example 1:

```
let fruits = ["Apple", "Banana"];  
fruits.push("Mango", "Orange");  
console.log(fruits);
```

Output: ['Apple', 'Banana', 'Mango', 'Orange']

We can even add elements with data types different from the ones in the array.

Example:

```
let fruits = ["Apple", "Banana"];  
fruits.push(5);  
console.log(fruits);
```

Output: ['Apple', 'Banana', 5]

5. shift():

Removes the first element from an array. This method changes the length of the array.

Example:

```
let fruits = ["Apple", "Banana", "Mango"];  
fruits.shift();  
console.log(fruits);
```

Output: ['Banana', 'Mango']

6. unshift():

Adds one or more elements to the beginning of an array. This method changes the length of the

array.

Example:

```
let fruits = ["Banana", "Mango"];
fruits.unshift(5, "Orange");
console.log(fruits);
```

Output: [5, 'Orange', 'Banana', 'Mango']

7. delete:

It is an operator used to remove a property from an object. When applied to an array, it deletes the element at the specified index, but it "does not change the length of the array, leaving undefined at the position of the deleted element".

Example:

```
let fruits = ["Apple", "Banana", "Mango"];
delete fruits[1];
console.log(fruits);
console.log(fruits.length);
```

Output:

```
[ 'Apple', undefined, 'Mango' ]
3
```

8. concat():

Example 1:

```
let teamA = ["Alice", "Bob"];
let teamB = ["Charlie", "David"];
let teamC = ["Eve", "Frank"];
let allTeams = teamA.concat(teamB, teamC);

console.log(allTeams);
```

Output: ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank']

Example 2: Concatenating Arrays with Values.

```
let numbers = [1, 2, 3];
let moreNumbers = numbers.concat(4, 5, [6, 7]);

console.log(moreNumbers);
```

Output: [1, 2, 3, 4, 5, 6, 7]

9. sort():

Sorts the elements of an array in place and returns the sorted array. By default, it sorts the elements as strings in ascending order.

Example 1:

```
let fruits = ["Banana", "Apple", "Mango", "Orange"];
fruits.sort();
console.log(fruits);
```

Output: ["Apple", "Banana", "Mango", "Orange"]

Example 2:

```
let numbers = [4, 2, 5, 1, 3];
numbers.sort();
console.log(numbers);
```

Output: [1, 2, 3, 4, 5]

10. splice():

It changes the contents of an array by removing, replacing, or adding elements. It modifies the original array.

Example 1: Removing elements

```
let fruits = ["Banana", "Apple", "Mango", "Orange"];
fruits.splice(1, 2);
console.log(fruits);
```

Output: ['Banana', 'Orange']

Example 2: Adding elements

```
let fruits = ["Banana", "Orange"];
fruits.splice(1, 0, "Apple", "Mango");
console.log(fruits);
```

Output: ["Banana", "Apple", "Mango", "Orange"]

Explanation of the above:

1: The index at which to start adding the new elements.
0: Indicates that no elements should be removed.
"Apple", "Mango": The elements to add to the array.

One more example for adding elements:

```
let fruits = ["Banana", "Orange"];
fruits.splice(1, 1, "Apple", "Mango");
console.log(fruits);
```

Output: ['Banana', 'Apple', 'Mango']

Explanation:

1: The index at which to start adding the new elements.
1: Indicates that one element, i.e, "Orange" should be removed.
"Apple", "Mango": The elements to add to the array.

Example 3: Replacing elements

```
let fruits = ["Banana", "Apple", "Mango", "Orange"];
```

```
fruits.splice(1, 2, "Grapes", "Pineapple");  
console.log(fruits);
```

Output: ["Banana", "Grapes", "Pineapple", "Orange"]

Explanation:

1: The index at which to start changing the array.

2: The number of elements to remove from the array, starting at index 1.

"Grapes", "Pineapple": The elements to add to the array, beginning at the start index. If 1

11. slice()

It returns a shallow copy of a portion of an array into a new array object. It does not modify the original array.

Example 1:

```
let fruits = ["Banana", "Apple", "Mango", "Orange"];  
fruits.slice(2, 4);  
console.log(fruits);
```

Output: ["Banana", "Apple", "Mango", "Orange"]

Explanation:

2: The index at which to begin extraction. The default is 0.

4: The index at which to end extraction. The default is the array's length. The extraction

Example 2: Copying the entire array.

```
let fruits = ["Banana", "Apple", "Mango", "Orange"];  
let allFruits = fruits.slice();  
console.log(allFruits);
```

Output: ["Banana", "Apple", "Mango", "Orange"]

12. reverse()

Example:


```
let fruits = ["Banana", "Apple", "Mango", "Orange"];  
fruits.reverse();  
console.log(fruits);
```

Output: ["Orange", "Mango", "Apple", "Banana"]