

JavaScript Functions :

1. `concat()` : This function is used to join two arrays. This does not change the original array but returns a new array as a result.

Syntax : `array1.concat(array2);`

Program :

```
1 var array1 = [1,2,3,4,5];
2 var array2 = ["Abhi","Akshat","Karthik","Rahul"];
3 var resultArray = array1.concat(array2);
4 console.log(resultArray);
```

Result :

```
Filter Output
Array(9) [ 1, 2, 3, 4, 5, "Abhi", "Akshat", "Karthik", "Rahul" ]
```

2. `every()` : This method is used to check if every element of the array passes the specified condition. It returns true if every element satisfies the condition or else it returns false.

Syntax : `array.every(Function(){});`

Program :

```
var array1 = [1,2,3,4,5];
var array2 = [6,7,8];

function check(item){
  return item<=5;
}

var result1 = array1.every(check);
var result2 = array2.every(check);

console.log(result1);
console.log(result2);
```

Output :

```
true
false
```

3. filter() : It is used to filter out elements from an array that passes the specified condition.

Syntax : array.filter(Function(){});

Program :

```
var array1 = [1,2,3,4,5];

// Return even numbers from array

function check(item){
  return ((item%2)==0);
}

var result = array1.filter(check);
console.log(result);
```

Output :

```
► Array [ 2, 4 ]
```

4. forEach() : It calls the specified function once for each and every element of the array.

Syntax : array.forEach(Function(){});

Program :

```
var array = [1,2,3,4,5];

array.forEach(function(item, index) {
  console.log('array['+index+'] : '+item);
});
```

Output :

```
array[0] : 1
array[1] : 2
array[2] : 3
array[3] : 4
array[4] : 5
```

5. `indexOf()` : This function is used to find the index of an element in the array. It return -1 if the element is not present.

Syntax : `array.indexOf(ele);`

Program :

```
var array = [1,2,3,4,5];  
  
console.log(array.indexOf(3));  
console.log(array.indexOf(9));
```

Output :

```
2  
-1
```

6. `join()` : This method is used to combine the elements of an array along with the seperator and it returns the resultant string.

Syntax : `array.join(seperator);`

Program :

```
var array = ["Abhijit","is","Good."];  
console.log(array.join(" "));
```

Output :

```
Abhijit is Good.
```

7. `lastIndexOf()` : This method as the name suggests returns the index of last occurrence of the element specified if present else return -1.

Syntax : `array.lastIndexOf(element);`

Program :

```
var array = [1,2,3,4,2];  
  
console.log(array.lastIndexOf(2));  
console.log(array.lastIndexOf(20));
```

Output :

```
4  
-1
```

8. `map()` : This method is used to create a new array with the results of the calling function.

Syntax : `array.map(function(){});`

Program :

```
var array = [1,2,3,4];  
  
var result = array.map(function (item){  
    return item*2;  
});  
  
console.log(result);
```

Output :

```
▶ Array(4) [ 2, 4, 6, 8 ]
```

9. pop() : This method removes the last element of the array and returns it.

Syntax : array.pop();

Program :

```
var array = [1,2,3,4];  
  
console.log(array.pop());  
console.log(array);
```

Output :

```
4  
▶ Array(3) [ 1, 2, 3 ]
```

10. push() : This method adds a new element at the end of the array and returns the length.

Syntax : array.push(element);

Program :

```
var array = [1,2,3,4];  
console.log(array.push(5));  
console.log(array);
```

Output :

```
5  
▶ Array(5) [ 1, 2, 3, 4, 5 ]
```

11. `reduce()` : This method reduces an array to a single value by executing the specified function on each element of the array(from left to right).

Syntax : `array.reduce(function (){ });`

Program :

```
var array = [1,2,3,4];
var result = 0;
result = array.reduce(function (result,item){
    return result + item;
});
console.log(result);
```

Output :

10

12. `reduceRight()` : This method reduces an array to a single value by executing the specified function on each element of the array(from right to left).

Syntax : `array.reduce(function (){ });`

Program :

```
var array = [1,2,3,4];
var result = 0;
result = array.reduceRight(function (result,item){
    return result + item;
});
console.log(result);
```

Output :

10

13. reverse() : This method reverses the contents of the array.

Syntax : array.reverse();

Program :

```
var array = [1,2,3,4];  
array.reverse();  
console.log(array);
```

Output :

```
► Array(4) [ 4, 3, 2, 1 ]
```

14. shift() : This method removes the first element of the array and return it.

Syntax : array.shift();

Program :

```
var array = [1,2,3,4];  
var result = array.shift();  
console.log(result);  
console.log(array);
```

Output :

```
1
```

```
► Array(3) [ 2, 3, 4 ]
```

15. unshift() : This method adds the element to the array and returns length.

Syntax : array.unshift(element);

Program :

```
var array = [1,2,3,4];  
var result = array.unshift(0);  
console.log(result);  
console.log(array);
```

Output :

```
5  
▶ Array(5) [ 0, 1, 2, 3, 4 ]
```

16. slice() : This method extracts the specified elements into a new array and returns it.

Syntax : array.slice(startIndex, endIndex);

Program :

```
var array = [1,2,3,4];  
var result = array.slice(0,2);  
console.log(result);
```

Output :

```
▶ Array [ 1, 2 ]
```


17. `some()` : This method checks if any of the elements of the array passes the specified condition. If any of the elements pass it returns true or else it returns false.

Syntax : `array.some(function (){});`

Program :

```
var array1 = [1,2,3,4,7,8];
var array2 = [10,11,12];

function check(item){
    return item<=5;
}

console.log(array1.some(check));
console.log(array2.some(check));
```

Output :

```
true
false
```

18. `sort()` : This method as the name suggests sorts the array and returns the sorted array. This method works well for strings by default.

Syntax : `array.sort();`

Program :

```
var array = ["Apple","Lenovo","Berry","Xiommi","Nokia","Motorolla"];
array.sort();
console.log(array);
```

Output :

```
► Array(6) [ "Apple", "Berry", "Lenovo", "Motorolla", "Nokia", "Xiommi" ]
```

19. `splice()` : This method adds/removes items to/from an array, and returns the removed item.

Syntax : `array.splice(startIndex, NumElementsToDelete, replacements[one or more]);`

Program :

```
var array = ["Apple", "Lenovo", "Berry", "Xiommi", "Nokia", "Motorolla"];
array.splice(2, 0, "Samsung", "Sony");
console.log(array);
var result = array.splice(2, 2, "Pixel");
console.log(result);
console.log(array);
```

Output :

```
▶ Array(8) [ "Apple", "Lenovo", "Samsung", "Sony", "Berry", "Xiommi", "Nokia", "Motorolla" ]
▶ Array [ "Samsung", "Sony" ]
▶ Array(7) [ "Apple", "Lenovo", "Pixel", "Berry", "Xiommi", "Nokia", "Motorolla" ]
```

20. toString() : This method converts numbers to string.

Syntax : num.toString();

Program :

```
var num = 5;
var stringNum = num.toString();
console.log(typeof(num));
console.log(typeof(stringNum));
console.log(num);
```

Output :

```
number
```

```
string
```

```
5
```

21. toSource() : This method represents the source code of an object.

Syntax : variable.toSource();

Program :

```
var num = 5;  
var string = "Abhijit";  
var float = 13.56;  
console.log(num.toString());  
console.log(string.toString());  
console.log(float.toString());
```

Output :

```
(new Number(5))  
(new String("Abhijit"))  
(new Number(13.56))
```

22. Difference between \n and \r.

The main difference between a line feed(\n) and carriage return(\r) is, line feed is used to move to one line forward whereas carriage return is used to move the cursor to the start of the line.