1) The **concat()** method is used to join two or more arrays.

2) The **copyWithin()** method copies array elements to another position in the array, overwriting the existing values.(won’t add new elements, size is same after the opertaion)   
array.copyWithin(target, start, end)  
target Required. The index position to copy the elements to  
start Optional. The index position to start copying elements from (default is 0)  
end Optional. The index position to stop copying elements from (default is array.length)

3) The **entries()** method returns an Array Iterator object with key/value pairs.

4) The **every()** method checks if all elements in an array pass a test (provided as a function).  
-- If it finds an array element where the function returns a *false* value, every() returns false (and does not check the remaining values)  
--If no false occur, every() returns true

5) The **fill()** method fills the specified elements in an array with a static value.  
array.fill(“value”, start, end)  
Value Required. The value to fill the array with  
start Optional. The index position to start copying elements from (default is 0)  
end Optional. The index position to stop copying elements from (default is array.length)

6) The **filter()** method creates an array filled with all array elements that pass a test (provided as a function).  
(the element which will pass the condition will be printed)

7) **push()** (add element at the end of the array)

8) **pop()** (removes the element from the end of an array)

9) let a =[1,2,3,4,5];  
undefined

a.**splice**(0,2);  
(2) [1, 2]

console.log(a);  
(3) [3, 4, 5]

a.**splice**(2,0,2.5,2.6); Second element = 0 defines not to delete any element from the array[]  
console.log(a);  
[3, 4, 2.5, 2.6, 5]  
undefined

a.**splice**(2,1,100,206); second element = 1 defines delete one element from array which is at position 2  
[2.5]

console.log(a);  
[3, 4, 100, 206, 2.6, 5]  
undefined

10) **indexOf()**  
specify the position of an element in the array  
return -1 , if the element is not found  
syntax: array.indexOf(item, start)

item : the item to search for  
start : from where to start(Optional)

11) The **map()** method creates a new array with the results of calling a function for every array element.  
tho the original array doesn’t get modified ,objects in the array gets modified  
e.g.  
var numbers = [4, 9, 16, 25];  
var x = numbers.**map**(Math.sqrt)  
document.getElementById("demo").innerHTML = x;

12) **filter()**filters the array to the value which we wantif particular element satisfies the condition it will be put into a new array.  
e.g  
arr = [1,2,3,4]  
newarr = arr.filter(a=>a%2===0)

13) **reduce()**in reduce we pass two arguments(accumulator and the current value)  
returns a number instead of an array  
reduce() method reduce the array to a single value  
benefit: With **reduce** you can filter and then map in a single pass

14) **from()**  
The Array.from() method returns an Array object from any object with a length property or an iterable object

15) **includes()**The includes() method determines whether an array contains a specified element.  
this method returns true if the array contains the element , and false if not

16) **isArray()**The isArray() method determines whether an object is an array.  
This function returns true if the object is an array, and false if not.

17) **join()**The join() method returns the array as a string.  
The elements will be separated by a specified separator. The default separator is comma (,).

18) **keys()**The keys() method returns an Array Iterator object with the keys of an array.

19) **length()**The length property sets or returns the number of elements in an array.

20) **reduceRight()**  
The reduceRight() method reduces the array to a single value.  
The reduceRight() method executes a provided function for each value of the array (from right-to-left).  
The return value of the function is stored in an accumulator (result/total)

21) **reverse()**  
The reverse() method reverses the order of the elements in an array.

22) **shift()**The shift() method removes the first item of an array.

23) **slice()**The slice() method returns the selected elements in an array, as a new array object.The slice() method selects the elements starting at the given start argument, and ends at, but does not include, the given end argument

sort(), some()