**1. push()**

Adds one or more elements to the end of an array and returns the new length of the array.

const fruits = ['apple', 'banana']; fruits.push('orange');

console.log(fruits); // ['apple', 'banana', 'orange']

**2. pop()**

Removes the last element from an array and returns that element.

const fruits = ['apple', 'banana', 'orange'];

const last = fruits.pop(); console.log(last); // 'orange' console.log(fruits); // ['apple', 'banana']

**3. shift()**

Removes the first element from an array and returns that removed element.

const fruits = ['apple', 'banana', 'orange'];

const first = fruits.shift(); console.log(first); // 'apple' console.log(fruits); // ['banana', 'orange']

**4. unshift()**

Adds one or more elements to the beginning of an array and returns the new length of the array.

const fruits = ['banana', 'orange'];

const newLength = fruits.unshift('apple'); console.log(fruits); // ['apple', 'banana', 'orange'] console.log(newLength); // 3

**5. forEach()**

Executes a provided function once for each array element.

const fruits = ['apple', 'banana', 'orange'];

fruits.forEach(fruit => console.log(fruit));

**6. map()**

Creates a new array populated with the results of calling a provided function on every element in the calling array.

const numbers = [1, 2, 3];

const squares = numbers.map(num => num \* num); console.log(squares); // [1, 4, 9]

**7. filter()**

Creates a new array with all elements that pass the test implemented by the provided function.

const numbers = [1, 2, 3, 4, 5];

const evenNumbers = numbers.filter(num => num % 2 === 0);

console.log(evenNumbers); // [2, 4]

**8. reduce()**

Executes a reducer function (that you provide) on each member of the array resulting in a single output value.

const numbers = [1, 2, 3, 4];

const sum = numbers.reduce((accumulator, currentValue) => accumulator + currentValue, 0); console.log(sum); // 10

**9. some()**

Tests whether at least one element in the array passes the test implemented by the provided function. Returns a Boolean.

const numbers = [1, 2, 3, 4]; const hasEvenNumber = numbers.some(num => num % 2 === 0); console.log(hasEvenNumber); // true

**10. every()**

Tests whether all elements in the array pass the test implemented by the provided function. Returns a Boolean.

const numbers = [2, 4, 6, 8];

const allEven = numbers.every(num => num % 2 === 0);

console.log(allEven); // true

**11. find()**

Returns the value of the first element in the provided array that satisfies the provided testing function.

const numbers = [5, 12, 8, 130, 44];

const found = numbers.find(num => num > 10);

console.log(found); // 12

**12. findIndex()**

Returns the index of the first element in the array that satisfies the provided testing function.

const numbers = [5, 12, 8, 130, 44];

const largeNumIndex = numbers.findIndex(num => num > 15); console.log(largeNumIndex); // 3

**13. slice()**

Returns a shallow copy of a portion of an array into a new array object selected from start to end (end not included).

const fruits = ['apple', 'banana', 'orange', 'mango'];

const citrus = fruits.slice(2, 3); console.log(citrus); // ['orange']

**14. splice()**

Changes the contents of an array by removing or replacing existing elements and/or adding new elements in place.

const fruits = ['apple', 'banana', 'orange', 'mango']; fruits.splice(2, 1, 'lemon', 'kiwi'); console.log(fruits); // ['apple', 'banana', 'lemon', 'kiwi', 'mango']

**15. concat()**

Used to merge two or more arrays. This method does not change the existing arrays but instead returns a new array.

const arr1 = ['a', 'b', 'c']; const arr2 = ['d', 'e', 'f']; const arr3 = arr1.concat(arr2); console.log(arr3); // ['a', 'b', 'c', 'd', 'e', 'f']

**16. join()**

Joins all elements of an array into a string.

const elements = ['Fire', 'Air', 'Water'];

const joined = elements.join();

console.log(joined); // "Fire,Air,Water"

console.log(elements.join(' + ')); // "Fire + Air + Water"

**17. reverse()**

Reverses an array in place. The first array element becomes the last, and the last array element becomes the first.

const array = [1, 2, 3];

array.reverse(); console.log(array); // [3, 2, 1]

**18. sort()**

Sorts the elements of an array in place and returns the sorted array.

const months = ['March', 'Jan', 'Feb', 'Dec'];

months.sort(); console.log(months); // ['Dec', 'Feb', 'Jan', 'March']