



JavaScript Array Methods

2nd edition

by
Ankush Bhagat



Array Creation & Modification Methods

1. Array.from()

Creates an array from an iterable or array-like object.



```
1 const str = 'hello';  
2 const arr = Array.from(str);  
3 console.log(arr); // ['h', 'e', 'l', 'l', 'o']
```

2. Array.isArray()

Checks if the provided value is an array.



```
1 console.log(Array.isArray([1, 2, 3])); // true  
2 console.log(Array.isArray('hello')); // false
```

3. Array.of()

Creates a new array instance with a variable number of arguments.



```
1 const arr = Array.of(1, 2, 3);  
2 console.log(arr); // [1, 2, 3]
```

4. concat()

Combines two or more arrays into a new array without modifying the original ones.



```
1 const arr1 = [1, 2];  
2 const arr2 = [3, 4];  
3 const combined = arr1.concat(arr2);  
4 console.log(combined); // [1, 2, 3, 4]
```

5. copyWithin()

Shallow copies part of an array to another location within the same array.



```
1 const arr = [1, 2, 3, 4, 5];  
2 arr.copyWithin(0, 3);  
3 console.log(arr); // [4, 5, 3, 4, 5]
```

6. fill()

Fills all elements of an array with a static value, from a start index to an end index.



```
1 const arr = [1, 2, 3];  
2 arr.fill(0);  
3 console.log(arr); // [0, 0, 0]
```

7. pop()

Removes and returns the last element of an array, reducing its length.



```
1  const arr = [1, 2, 3];
2  const last = arr.pop();
3  console.log(last); // 3
4  console.log(arr);  // [1, 2]
```

8. push()

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



```
1  const arr = [1, 2];
2  arr.push(3);
3  console.log(arr); // [1, 2, 3]
```

9. shift()

Removes and returns the first element from an array, shifting all other elements.



```
1  const arr = [1, 2, 3];
2  const first = arr.shift();
3  console.log(first); // 1
4  console.log(arr);  // [2, 3]
```

10. unshift()

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



```
1 const arr = [2, 3];  
2 arr.unshift(1);  
3 console.log(arr); // [1, 2, 3]
```

11. reverse()

Reverses the order of the elements in the array.



```
1 const arr = [1, 2, 3];  
2 arr.reverse();  
3 console.log(arr); // [3, 2, 1]
```

12. splice()

Adds, removes, or replaces elements in an array by modifying the original array.



```
1 const arr = [1, 2, 3, 4];  
2 arr.splice(1, 2);  
3 console.log(arr); // [1, 4]
```

13. sort()

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



```
1 let arr = new Array(3); // [empty × 3]
```

14. Array()

Array constructor to create arrays.



```
1 let arr = new Array(3); // [empty × 3]
```


Iteration & Access Methods

15. at()

Returns the element at the given index (can be negative for reverse indexing).



```
1 const arr = [1, 2, 3];  
2 console.log(arr.at(-1)); // 3
```

16. forEach()

Executes a provided function once for each array element.



```
1 const arr = [1, 2, 3];  
2 arr.forEach((num) => console.log(num)); // 1, 2, 3
```

17. map()

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



```
1 const arr = [1, 2, 3];  
2 const newArr = arr.map(num => num * 2);  
3 console.log(newArr); // [2, 4, 6]
```

18. filter()

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



```
1 const arr = [1, 2, 3, 4];
2 const even = arr.filter(num => num % 2 === 0);
3 console.log(even); // [2, 4]
```

19. reduce()

Reduces the array to a single value by applying a function against each element.



```
1 const arr = [1, 2, 3];
2 const sum = arr.reduce((acc, val) => acc + val, 0);
3 console.log(sum); // 6
```

20. find()

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



```
1 const arr = [1, 2, 3, 4];
2 const found = arr.find(num => num > 2);
3 console.log(found); // 3
```


21. findIndex()

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



```
1 const arr = [1, 2, 3, 4];  
2 const index = arr.findIndex(num => num > 2);  
3 console.log(index); // 2
```

22. findLast()

Returns the last element that satisfies the provided testing function.



```
1 const arr = [1, 2, 3, 4];  
2 const foundLast = arr.findLast(num => num > 2);  
3 console.log(foundLast); // 4
```

23. findLastIndex()

Returns the index of the last element that satisfies the provided testing function.



```
1 const arr = [1, 2, 3, 4];  
2 const indexLast = arr.findLastIndex(num => num > 2);  
3 console.log(indexLast); // 3
```

24. every()

Tests whether all elements in the array pass the provided function.



```
1 const arr = [1, 2, 3];  
2 const allLessThan4 = arr.every(num => num < 4);  
3 console.log(allLessThan4); // true
```

25. some()

Tests whether at least one element in the array passes the provided function.



```
1 const arr = [1, 2, 3];  
2 const hasGreaterThan2 = arr.some(num => num > 2);  
3 console.log(hasGreaterThan2); // true
```

26. includes()

Determines whether an array includes a certain value.



```
1 const arr = [1, 2, 3];  
2 console.log(arr.includes(2)); // true
```

27. indexOf()

Returns the first index at which a given element can be found.



```
1 const arr = [1, 2, 3, 2];  
2 console.log(arr.indexOf(2)); // 1
```

28. lastIndexOf()

Returns the last index at which a given element can be found.



```
1 const arr = [1, 2, 3, 2];  
2 console.log(arr.lastIndexOf(2)); // 3
```

Array Transformation Methods

29. join()

Joins all elements of an array into a string, separated by a specified separator.



```
1 const arr = [1, 2, 3];  
2 console.log(arr.join('-')); // "1-2-3"
```

30. toString()

Converts an array to a string, with elements separated by commas.



```
1 const arr = [1, 2, 3];  
2 console.log(arr.toString()); // "1,2,3"
```

Array Access & Utility Methods

31. slice()

Returns a shallow copy of a portion of an array.



```
1 let arr = [1, 2, 3, 4];  
2 let sliced = arr.slice(1, 3); // [2, 3]
```

32. entries()

Returns an iterator with key/value pairs.



```
1 let arr = ['a', 'b', 'c'];  
2 for (let [index, value] of arr.entries()) {  
3   console.log(index, value);  
4 }  
5 // 0 "a"  
6 // 1 "b"  
7 // 2 "c"
```

33. keys()

Returns an iterator containing the keys for each index.



```
1 let arr = ['a', 'b', 'c'];
2 for (let key of arr.keys()) {
3     console.log(key);
4 }
5 // 0
6 // 1
7 // 2
```

34. values()

Returns an iterator containing the values for each index.



```
1 let arr = ['a', 'b', 'c'];
2 for (let value of arr.values()) {
3     console.log(value);
4 }
5 // "a"
6 // "b"
7 // "c"
```


35. flat()

Flattens sub-array elements into the main array.



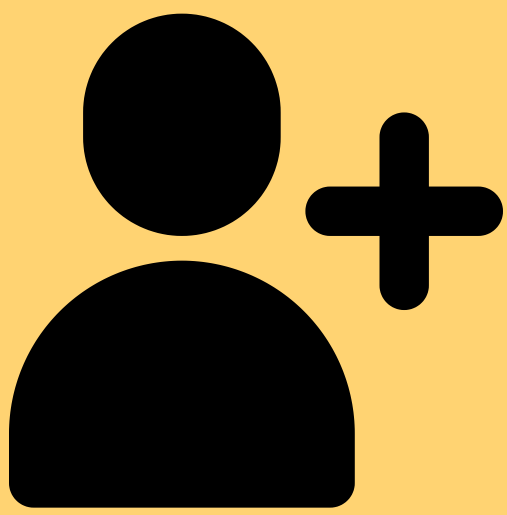
```
1 let arr = [1, [2, 3], [4, [5]]];  
2 console.log(arr.flat(2)); // [1, 2, 3, 4, 5]
```

36. flatMap()

Maps each element using a function and flattens the result into a new array.



```
1 let arr = [1, 2, 3];  
2 let mapped = arr.flatMap(e => [e * 2]); // [2, 4, 6]
```



Follow

for

more

