**JavaScript Array Methods with Examples**

**1. push() → Adds element(s) to the end**

let arr = [1, 2];

arr.push(3, 4);

console.log(arr); // [1, 2, 3, 4]

**2. pop() → Removes last element**

let arr = [1, 2, 3];

arr.pop();

console.log(arr); // [1, 2]

**3. shift() → Removes first element**

let arr = [10, 20, 30];

arr.shift();

console.log(arr); // [20, 30]

**4. unshift() → Adds element(s) at the beginning**

let arr = [2, 3];

arr.unshift(1);

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

**5. concat() → Merges arrays**

let a = [1, 2];

let b = [3, 4];

console.log(a.concat(b)); // [1, 2, 3, 4]

**6. join() → Converts array to string**

let arr = ["red", "green", "blue"];

console.log(arr.join("-")); // red-green-blue

**7. slice() → Returns part of array (non-destructive)**

let arr = [10, 20, 30, 40];

console.log(arr.slice(1, 3)); // [20, 30]

console.log(arr.slice(-2)); // [30, 40]

**8. splice() → Adds/removes items (destructive)**

let arr = [1, 2, 3, 4];

arr.splice(1, 2, 99);

console.log(arr); // [1, 99, 4]

**9. indexOf() → First occurrence index**

let arr = [5, 10, 15, 10];

console.log(arr.indexOf(10)); // 1

**10. lastIndexOf() → Last occurrence index**

let arr = [5, 10, 15, 10];

console.log(arr.lastIndexOf(10)); // 3

**11. includes() → Checks if value exists**

let arr = [1, 2, 3];

console.log(arr.includes(2)); // true

console.log(arr.includes(5)); // false

**12. forEach() → Iterates through array (no return)**

[1, 2, 3].forEach(num => console.log(num \* 2));

// 2, 4, 6

**13. map() → Creates new array with transformed values**

let arr = [1, 2, 3];

let doubled = arr.map(n => n \* 2);

console.log(doubled); // [2, 4, 6]

**14. filter() → Returns array of matching elements**

let arr = [5, 10, 15, 20];

let result = arr.filter(n => n > 10);

console.log(result); // [15, 20]

**15. reduce() → Reduces array to a single value**

let arr = [1, 2, 3, 4];

let sum = arr.reduce((acc, curr) => acc + curr, 0);

console.log(sum); // 10

**16. reduceRight() → Like reduce, but right-to-left**

let arr = ["a", "b", "c"];

let res = arr.reduceRight((acc, curr) => acc + curr);

console.log(res); // "cba"

**17. every() → Checks if *all* elements satisfy condition**

let arr = [2, 4, 6];

console.log(arr.every(n => n % 2 === 0)); // true

**18. some() → Checks if *at least one* element satisfies condition**

let arr = [1, 3, 5, 6];

console.log(arr.some(n => n % 2 === 0)); // true

**19. find() → Returns first matching element**

let arr = [10, 20, 30];

console.log(arr.find(n => n > 15)); // 20

**20. findIndex() → Returns index of first match**

let arr = [10, 20, 30];

console.log(arr.findIndex(n => n > 15)); // 1

**21. sort() → Sorts array (default: as strings)**

let arr = [100, 25, 3];

console.log(arr.sort()); // [100, 25, 3] ❌ (string sort)

console.log(arr.sort((a, b) => a - b)); // [3, 25, 100] ✅

**22. reverse() → Reverses array**

let arr = [1, 2, 3];

arr.reverse();

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

**23. flat() → Flattens nested arrays**

let arr = [1, [2, 3], [4, [5]]];

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

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

**24. flatMap() → map() + flat(1)**

let arr = [1, 2, 3];

console.log(arr.flatMap(x => [x, x \* 2]));

// [1, 2, 2, 4, 3, 6]

**25. fill() → Fills array with value**

let arr = [1, 2, 3, 4];

arr.fill(0, 1, 3);

console.log(arr); // [1, 0, 0, 4]

**26. copyWithin() → Copies part of array within itself**

let arr = [1, 2, 3, 4, 5];

arr.copyWithin(0, 3);

console.log(arr); // [4, 5, 3, 4, 5]

**27. keys(), values(), entries() → Iterators**

let arr = ["a", "b", "c"];

console.log([...arr.keys()]); // [0, 1, 2]

console.log([...arr.values()]); // ["a", "b", "c"]

console.log([...arr.entries()]); // [[0,"a"], [1,"b"], [2,"c"]]