

JavaScript-এ **Array Functions** হল বিল্ট-ইন মেথড, যা অ্যারের সাথে বিভিন্ন ধরনের কার্য সম্পাদন করতে সাহায্য করে। এগুলো ডেটা ম্যানিপুলেশনের জন্য খুবই কার্যকর। নিচে বিভিন্ন ধরনের গুরুত্বপূর্ণ Array Functions এর বর্ণনা দেওয়া হলো:

1. Traversal & Access:

forEach():

- অ্যারেতে প্রতিটি আইটেমের উপর একটি ফাংশন চালায়।

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

map():

- অ্যারেতে প্রতিটি আইটেমে একটি ফাংশন প্রয়োগ করে নতুন অ্যারে তৈরি করে।

```
const numbers = [1, 2, 3];
const squared = numbers.map(num => num ** 2);
console.log(squared); // Output: [1, 4, 9]
```

2. Filtering & Searching:

filter():

- নির্দিষ্ট শর্ত মেনে চলা আইটেমগুলো নিয়ে নতুন অ্যারে তৈরি করে।

```
const numbers = [1, 2, 3, 4, 5];
const evens = numbers.filter(num => num % 2 === 0);
console.log(evens); // Output: [2, 4]
```

find():

- শর্ত পূরণ করা প্রথম আইটেমটি রিটার্ন করে।

```
const numbers = [1, 2, 3, 4];
const firstEven = numbers.find(num => num % 2 === 0);
console.log(firstEven); // Output: 2
```

findIndex():

- শর্ত পূরণ করা প্রথম আইটেমের ইনডেক্স রিটার্ন করে।

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

includes():

- অ্যারেতে কোনো নির্দিষ্ট মান আছে কিনা তা যাচাই করে।

```
const fruits = ["apple", "banana"];  
console.log(fruits.includes("banana")); // Output: true
```

3. Reduction ও Aggregation:

reduce():

- অ্যারের সব আইটেমকে একত্রিত করে একটি মান রিটার্ন করে।

```
const numbers = [1, 2, 3];  
const sum = numbers.reduce((acc, num) => acc + num, 0);  
console.log(sum); // Output: 6
```

reduceRight():

- `reduce()` এর মতোই কাজ করে, কিন্তু ডানদিক থেকে শুরু হয়।

```
const numbers = [1, 2, 3];  
const sum = numbers.reduceRight((acc, num) => acc + num, 0);  
console.log(sum); // Output: 6
```

4. Adding, Removing, ও Modifying:

push():

- অ্যারের শেষে নতুন আইটেম যোগ করে।

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

pop():

- অ্যারের শেষ আইটেম সরিয়ে দেয়।

```
const numbers = [1, 2, 3];  
numbers.pop();  
console.log(numbers); // Output: [1, 2]
```

shift():

- অ্যারের প্রথম আইটেম সরিয়ে দেয়।

```
const numbers = [1, 2, 3];  
numbers.shift();  
console.log(numbers); // Output: [2, 3]
```

unshift():

- অ্যারের শুরুতে নতুন আইটেম যোগ করে।

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

splice():

- নির্দিষ্ট ইনডেক্স থেকে আইটেম যোগ, সরানো বা পরিবর্তন করে।

```
const numbers = [1, 2, 3];  
numbers.splice(1, 1, 5);  
console.log(numbers); // Output: [1, 5, 3]
```

5. Sorting & Reversing:

sort():

- অ্যারের আইটেমগুলো সাজায়।

```
const numbers = [3, 1, 2];  
numbers.sort();  
console.log(numbers); // Output: [1, 2, 3]
```

reverse():

- অ্যারের আইটেমগুলো উল্টো করে।

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

6. Combining & Splitting:

concat():

- একাধিক অ্যারে একত্রিত করে।

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

slice():

- অ্যারের একটি অংশ কপি করে নতুন অ্যারে তৈরি করে।

```
const numbers = [1, 2, 3, 4];
const sliced = numbers.slice(1, 3);
console.log(sliced); // Output: [2, 3]
```

join():

- অ্যারের সব আইটেমকে একটি স্ট্রিংয়ে যোগ করে।

```
const words = ["Hello", "World"];
const sentence = words.join(" ");
console.log(sentence); // Output: "Hello World"
```

7. Checking & Iteration:

some():

- অ্যারের কোনো একটি আইটেম শর্ত পূরণ করলে `true` রিটার্ন করে।

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

every():

- অ্যারের সব আইটেম শর্ত পূরণ করলে `true` রিটার্ন করে।

```
const numbers = [1, 2, 3];  
console.log(numbers.every(num => num > 0)); // Output: true
```