! Here are some examples demonstrating the usage of TypeScript array methods in various scenarios:

1. Filtering Even Numbers:

```typescript
let numbers: number[] = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
let evenNumbers = numbers.filter(num => num % 2 === 0); // [2, 4, 6, 8, 10]
```

2. Summing Array Elements:

```typescript
let numbers: number[] = [1, 2, 3, 4, 5];
let sum = numbers.reduce((acc, curr) => acc + curr, 0); // 15
```

3. Finding Maximum Value:

```typescript
let numbers: number[] = [3, 7, 2, 9, 5];
let max = numbers.reduce((acc, curr) => Math.max(acc, curr)); // 9
```

4. Checking for Presence of a Value:

```typescript
let numbers: number[] = [1, 2, 3, 4, 5];
let hasThree = numbers.includes(3); // true
```

5. Mapping Array to New Values:

```typescript
let numbers: number[] = [1, 2, 3, 4, 5];
let squaredNumbers = numbers.map(num => num * num); // [1, 4, 9, 16, 25]
```

6. Removing Duplicates:

```typescript
let numbers: number[] = [1, 2, 2, 3, 3, 3, 4, 5];
let uniqueNumbers = Array.from(new Set(numbers)); // [1, 2, 3, 4, 5]
```

7. Concatenating Arrays:

```typescript
let arr1: number[] = [1, 2, 3];
let arr2: number[] = [4, 5, 6];
let concatenatedArray = arr1.concat(arr2); // [1, 2, 3, 4, 5, 6]
```

8. Splitting String into Array:

```typescript
let sentence = "Hello, world!";
let words = sentence.split(" "); // ["Hello,", "world!"]
```

9. Checking Array Emptiness:

```typescript
let emptyArray: number[] = [];
let isEmpty = emptyArray.length === 0; // true
```

10. Reversing Array:

```typescript
let numbers: number[] = [1, 2, 3, 4, 5];
numbers.reverse(); // numbers is now [5, 4, 3, 2, 1]
```