**.map()**

The .map() function creates a new array by applying a function to each element of an existing array

**Examples :**

**1.Add 5 to each number in an Array**

```
let numbers = [10, 20, 30, 40];
let increasedNumbers = numbers.map(num => num + 5);
console.log(increasedNumbers); // Output: [15, 25, 35, 45]
```

**Explanation :** The .map() function adds 5 to each number in the numbers array, resulting in a new array of increased numbers.

**2.Suffix to a word**

```
let names = ["Alice", "Bob", "Charlie"];
let greetings = names.map(name => "Hello, " + name + "!");
console.log(greetings); // Output: ["Hello, Alice!", "Hello, Bob!", "Hello, Charlie!"]
```

**Explanation :** The map() function adds "ing" to each word, resulting in a new array actionWords with the modified words.

**3. Append “ing” to Each Word**

```

let words = ["read", "write", "code"];
let actionWords = words.map(word => word + "ing");
console.log(actionWords); // Output: ["reading", "writing", "coding"]
```

**Explanation:** The map() function adds "ing" to each word, resulting in a new array actionWords with the modified words.

**.reduce() :**
The .reduce() function combines all the elements of an array into a single value by processing each element according to a specified function.

**1.Find the Maximum Number in an Array**

```
let numbers = [10, 50, 25, 80, 45];
let maxNumber = numbers.reduce((max, num) => num > max ? num : max, numbers[0]);
console.log(maxNumber); // Output: 80
```

**Explanation :** The .reduce() function finds the maximum number in the numbers array by comparing each number to the current maximum.

**2.Create a Sentence from Array Elements**

```
let words = ["This", "is", "a", "sentence"];
let sentence = words.reduce((acc, word) => acc + " " + word);
console.log(sentence); // Output: "This is a sentence"
```

**Explanation :** The .reduce() function concatenates the words in the words array into a single sentence, with each word separated by a space.

**3. Count the Number of Elements**

```
let numbers = [1, 2, 3, 4, 5];
let count = numbers.reduce((acc, num) => acc + 1, 0);
console.log(count); // Output: 5
```

**Explanation :** The .reduce() function counts the number of elements in the numbers array by incrementing the accumulator by 1 for each element.

**.filter()**

The .filter() function in JavaScript is used to create a new array containing only the elements that meet a certain condition.

**Examples :**

**1. Filter Words Containing the Letter 'a'**

```
let words = ["apple", "banana", "cherry", "date"];
let wordsWithA = words.filter(word => word.includes('a'));
console.log(wordsWithA); // Output: ["apple", "banana", "date"]
```

**Explanation :** The .filter() function creates a new array with words from the words array that contain the letter 'a'.

**2.Filter Numbers Divisible by 3**

```
let numbers = [3, 5, 9, 12, 14];
let divisibleByThree = numbers.filter(num => num % 3 === 0);
console.log(divisibleByThree); // Output: [3, 9, 12]
```

**Explanation :** The .filter() function creates a new array with numbers from the numbers array that are divisible by 3.

**3.Filter Out Negative Numbers**

```
let numbers = [-1, -5, 3, 7, -2];
let nonNegativeNumbers = numbers.filter(num => num >= 0);
console.log(nonNegativeNumbers); // Output: [3, 7]
```

**Explanation :** The .filter() function creates a new array with numbers from the numbers array that are zero or positive.