1. forEach():

The forEach method is useful for iterating over array elements and performing operations on them. It provides a simple and readable way to process each element in an array without the need for managing loop counters explicitly.

```
Example:

const numbers = [1, 2, 3, 4, 5];
const doubledNumbers = [];

numbers.forEach((number) => {
    doubledNumbers.push(number * 2);
});

console.log(doubledNumbers);

Output: [2, 4, 6, 8, 10]
```

2. Array.from():

It creates a new, shallow-copied array instance from an array-like or iterable object, say string.

```
Example:

const str = 'hello';
const arr = Array.from(str);
console.log(arr);

Output: ['h', 'e', 'l', 'l', 'o']
```

3. map():

It creates a new array populated with the results of calling a provided function on every element in the calling array.

```
Example:

const numbers = [1, 2, 3, 4];
const doubled = numbers.map(x => x * 2);
```

```
console.log(doubled);

Output: [2, 4, 6, 8]
```

4. filter():

It creates a new array with all elements that pass the test implemented by the provided function.

```
Example:

const numbers = [1, 2, 3, 4, 5, 6];
const evenNumbers = numbers.filter(x => x % 2 === 0);
console.log(evenNumbers);

Output: [2, 4, 6]
```

5. reduce():

It executes a reducer function (that you provide) on each element of the array, resulting in a single output value.

```
Example:

const numbers = [1, 2, 3, 4, 5];

const sum = numbers.reduce((acc, x) => acc + x, 0);

console.log(sum);

Output: 15


Explanation:

Here acc is used to store the sum of the array and it's default value is set to 0.
x is each element of the array.
```

```
Factorial of a number ( Done by me ):

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

```
let prod = arr.reduce(function(acc, x){
    return acc*x;
}, 1);

console.log(prod);
```

Above code can also be written as:

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

let prod = arr.reduce((acc, x) => acc*x, 1);

console.log(prod);
```

Output: 120