**ARRAY ITERATION METHODS**

In JavaScript, array iteration methods provide powerful and convenient ways to operate on arrays. Here’s an overview of some of the most commonly used methods: **forEach**, **map**, **filter**, and **reduce**.

**1. forEach**

The **forEach** method executes a provided function once for each array element. It’s used for performing operations on each element of the array but does not return a new array.

**Syntax:**

array.forEach(function(element, index, array) {

// your code here });

**Example:**

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

numbers.forEach(function(number) {

console.log(number \* 2); }); // Output: 2, 4, 6, 8, 10

**2. map**

The **map** method creates a new array populated with the results of calling a provided function on every element in the calling array. It returns a new array without modifying the original array.

**Syntax:**

const newArray = array.map(function(element, index, array) {

// return value });

**Example:**

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

const doubled = numbers.map(function(number) {

return number \* 2; });

console.log(doubled); // Output: [2, 4, 6, 8, 10]

**3. filter**

The **filter** method creates a new array with all elements that pass the test implemented by the provided function. It returns a new array without modifying the original array.

**Syntax:**

const newArray = array.filter(function(element, index, array) {

// return boolean });

**Example:**

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

const evenNumbers = numbers.filter(function(number) {

return number % 2 === 0; });

console.log(evenNumbers); // Output: [2, 4]

**4. reduce**

The **reduce** method executes a reducer function (that you provide) on each element of the array, resulting in a single output value. It’s often used for accumulating values.

**Syntax:**

const result = array.reduce(function(accumulator, currentValue, index, array) {

// return updated accumulator }, initialValue);

**Example:**

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

const sum = numbers.reduce(function(accumulator, currentValue) { return accumulator + currentValue;

}, 0);

console.log(sum); // Output: 15

**Summary**

* **forEach**: Executes a provided function once for each array element. Does not return a new array.
* **map**: Creates a new array with the results of calling a provided function on every element. Returns a new array.
* **filter**: Creates a new array with all elements that pass the test implemented by the provided function. Returns a new array.
* **reduce**: Executes a reducer function on each element of the array, resulting in a single output value.

Each method is designed for different use cases, and choosing the right one depends on what you intend to do with the array elements.