ANSWER 7-

‘map’ and ‘forEach’ are both array methods in JavaScript used to iterate over arrays and perform actions on each element. However, they have some differences in terms of their return values and how they operate:

Return Value: ‘map’ returns a new array containing the results of applying a function to each element of the original array. It creates a new array by transforming each element based on the provided function and returns that new array. ‘forEach’ does not return anything. It simply iterates over each element of the array and executes a provided function for each element.

Usage and Purpose: ‘map’ is typically used when you want to transform an array into another array with modified elements. It's useful when you want to perform a specific operation on each element and collect the results in a new array. ‘forEach’ is used when you want to iterate over an array and perform some action or side effect for each element, such as logging values, modifying external variables, or triggering other functions. It's useful when you don't need to create a new array but want to perform a task on each element.

Modifying the Original Array: ‘map’ does not modify the original array. Instead, it creates a new array with transformed elements, leaving the original array unchanged. ‘forEach’ also does not modify the original array. It simply iterates over the elements and performs an action on each one.

Here's an example to illustrate the difference:

```
const numbers = [1, 2, 3];

// Using map
```

```
 const doubled = numbers.map(num => num * 2);
console.log(doubled);

// Output: [2, 4, 6]

// Using forEach

 numbers.forEach(num => { const doubledNum = num * 2;
console.log(doubledNum); // Output: 2, 4, 6 });
console.log(numbers);

// Output: [1, 2, 3] (unchanged)
```

In summary, ‘map’ is used to transform an array and create a new array with the results, while ‘forEach’ is used to iterate over an array and perform an action on each element without creating a new array.