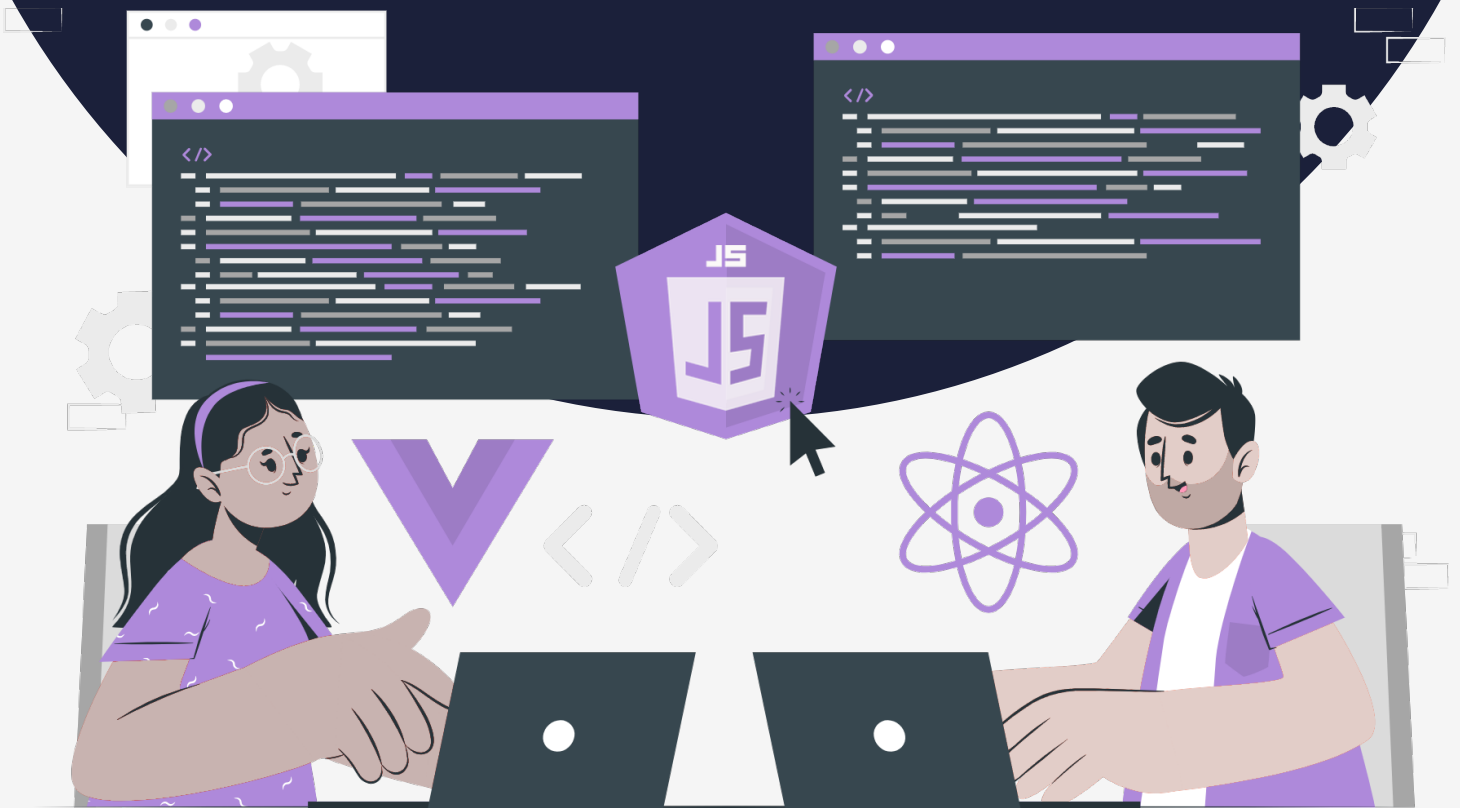# Lesson:

reduce

Topics Covered

1. Introduction.
2. Features of reduce.
3. Syntax.
4. Implementation.

As the name suggests the reduce method reduces the array values to a single value. The reduce() method runs a reducer function on each array item and returns a single output result. The input array can have numbers, strings, or an object.

Features of reduce:

1. The reducer function passed will be applied to all the items in the array.
2. The result will be a single value accumulated on passing all the array items to the reducer function.
3. The reduce method is not valid for empty arrays.
4. The reduce method doesn't change the original array on which the reduce method is applied.

Syntax:

```
// Syntax
array.reduce(function(accumulator, currentValue, currentIndex, array), initialValue)
```

The parameters accepted:

1. accumulator: It accumulates the initial value and reducer function's return values.
2. currentValue: It is a required parameter and is used to specify the value of the current element.
3. currentIndex: It is an optional parameter and is used to specify the array index of the current element.
4. array: It is an optional parameter and is used to specify the array the current element belongs to.
5. initialValue: It is used to specify the value to be passed to the function as the initial value.

Let's look at the implementation of reduce method. The simplest example to demonstrate the use of reduce function is by finding the sum of all elements in an array.

```
// Find the sum of all elements in an array?
// Given Array

let arr = [1, 2, 44, 67, 89]

// Find the sum of all elements in an array?

// Given Array
let arr = [1, 2, 44, 67, 89];

// Apply reduce method
let result = arr.reduce((acc, curr) ⇒ acc + curr, 0);

// Print the result to the console.console.log(result);
```

We are passing in a callback function as the reduce function (acc, curr) => acc + curr, which takes in the accumulator and current value as arguments, and adds the current value to the accumulator. We have also passed in the initial value as 0, which is the starting value of the accumulator.

Note: If the initial value is not passed to the reduce method, the first item in the sequence is used as the initial value and the operation starts from the second item in the sequence. If the sequence is empty, a TypeError is raised.

The reduce method iterates through the array and for each element, it applies the callback function and updates the accumulator, adding the current value to the accumulator and storing the result in the accumulator.

After iterating through the entire array, the final value of the accumulator is returned, which is the sum of all elements in the array