JavaScript
map( ) vs forEach( )

map( ) method

The map( ) method is used for creating a new array from the existing one by applying a function to each one of the elements of the first array.

```
let nums = [2, 4, 5, 6];
let multNum = nums.map(num => num * num);
console.log(multNum);          //  Output: [ 4, 16, 25, 36 ]
```

forEach( ) method

The forEach( ) method receives a function as an argument and executes it once for each array element. Same like map( ) method.

However, instead of returning a new array like map, it returns undefined.

```
let nums = [2, 4, 5, 6];
let multNum = nums.forEach(num => console.log(num * num));
// output: 4 16 25 36
```

The returning value

The first difference between map( ) and forEach( ) is the returning value.

The map returns a new array with the transformed elements whereas forEach method returns undefined.

```
let nums = [2, 4, 5, 6];
let multNum = nums.map(num => num * num);
console.log(multNum);          //  Output: [ 4, 16, 25, 36 ]
```

```
let nums = [2, 4, 5, 6];
let multNum = nums.forEach(num => num * num);
console.log(multNum);    // output: undefined
```

Ability to chain

The second difference between these array methods is the fact that map() is chainable. This means that we can attach reduce (), sort(), filter() and so on after performing a map() method on an array.

That's something we can't do with forEach() because, it returns undefined.

```
let nums = [2, 4, 5, 6];
let multNum = nums
  .map((num) => num * num)
  .reduce((acc, curr) => acc + curr);
console.log(multNum);
// output: 81
```

```
let nums = [2, 4, 5, 6];
let multNum = nums
  .forEach((num) => num * num)
  .reduce((acc, curr) => acc + curr);
console.log(multNum);
// TypeError: Cannot read properties of undefined (reading 'reduce')
```

When to use what

The choice between map() and forEach() will completely depend on the use case.

If we need to change, alternate, or use the data, we should pick a map(), because as we know that it returns a new array with the transformed data.

But, suppose we don't need the returned array, so we can skip the map() and will use forEach() or even a for loop.

FOLLOW