**Helper methods built into JavaScript Array Object**

1. forEach
2. filter
3. map
4. find
5. reduce
6. every and some

**forEach:**

var colors = [‘red’,’white’,’blue’];

**Traditional way of looping through an array**

*for(var i=0;i<colors.length;i++){*

*console.log(colors[i]);*

*}*

**Looping an array using forEach method**

*colors.forEach(function(color){*

*console.log(color);*

*})*

Here forEach colors element, we are passing each element to an anonymous function called **iterator function** and the function is doing something on each element.

**map:**

This is the most widely used array helper around. Below code sample has two arrays variables and one is empty. We want to load second array with 2 multiplied values of first array numbers. We should not use same numbers array to write new values because it is **immutable**, hence we are using a different array.

*var numbers = [1,2,3,4,5];*

*var twoMultipliedNumbers =[];*

*forEach.numbers(function(number){*

*squaredNumber.push(number\*2);*

*});*

But if we use map method of array we don’t need to create second array and push the values.

*var maptwoMultipliedValues = numbers.map(function(number){*

***return*** *(number \*2);*

*});*

Here important thing is iteration function should take each number and **return** new transformed value.

filter:

Anytime we want to filter any type of list then we use filter method of array object.

var product = [

{name:’celery’,type:’vegetable’},

{name:’spinach’,type:’vegetable’},

{name:’orange’,type:’fruit’},

{name:’chicken’,type:’meat’},

];

var fruitsList=[];

products.forEach(function(product){

if(product.type==’fruit’){

fruitsList.push(product);

}

})