﻿Functional Programming in JavaScript

Higher Order Functions:

These functions are values like any other values that is taken as parameters for other functions and these functions are also called object of array.

For example: Filter, Map, Reduce, Sort etc.

Map: Map is another higher order function just like filter. It takes the each item as parameter on callback/directly of the array and we can do something with it for return without changing the original array.

Syntax

array.map(function(currentValue, index, arr), currentValue)

Example 01:

employees: [

{"name":"Bablu Ahmed", "title":"Programmer"},

{"name":"Abdul Kuddus", "title":"Vue JS Expert"},

{"name":"Sakibur Rahman", "title":"Marketing Manager"},

]

Let’s try first using for loop as follows:

var names = [];

for (var i = 0; i<employees.length; i++)

{

names.push(employees[i].name);

}

console.log(names);

output:

["Bablu Ahmed", "Abdul Kuddus", "Sakibur Rahman"]

But we can do exact using Map and so much less code:

var names = animals.map(functions(animal){

return animal.name;

});

In arrow function:

var names = animals.map((animal)=> return animal.name);

Example 02:

var numbers = [4, 9, 16, 25];

function myFunction() {

x = document.getElementById("demo")

x.innerHTML = numbers.map(Math.sqrt);

}

Output:

2,3,4,5

Some other functions:

Split, replace, slice, filter,

Split:

slice:

**The Spread (…)**

jj