**JAVASCRIPT**

**SPLICE** :---**Functional Programming: Remove Elements from an Array Using slice Instead of splice---**

when you want to remove items and keep the rest of the array. JavaScript offers the splice method for this, which takes arguments for the index of where to start removing items, then the number of items to remove. If the second argument is not provided, the default is to remove items through the end. However, the splice method mutates the original array it is called on.

**let arr = ['foo', 'bar', 10, 'qux'];**

***// arr.splice(<index>, <steps>, [elements ...]);***

**arr.splice(1, 1);           *// Removes 1 item at index 1***

***// => ['foo', 10, 'qux']***

**arr.splice(2, 1, 'tmp');    *// Replaces 1 item at index 2 with 'tmp'***

***// => ['foo', 10, 'tmp']***

**arr.splice(0, 1, 'x', 'y'); *// Inserts 'x' and 'y' replacing 1 item at index 0***

***// => ['x', 'y', 10, 'tmp']***

**Concat : ---Concat offers a way to add new items to the end of an array without any mutating side effects.--**

**---The variable watchList holds an array of objects with information on several movies. Use reduce to find the average IMDB rating of the movies directed by Christopher Nolan.--**

**var listOfMovies = watchList.filter(item=>item.Director === "Christopher Nolan")**

**var averageRating = listOfMovies.map(item=>item.imdbRating)**

**.reduce((a,b)=> parseFloat(a)+parseFloat(b))/listOfMovies.length;**

**---Functional Programming: Sort an Array Alphabetically using the sort Method--**

**["B", "O", "A", "M"].sort();**

**---Combine an Array into a String Using the join Method---**

**"M a n g o".split(/\W/).join('')**

**Every ---The every method works with arrays to check if every element passes a particular test. It returns a Boolean value - true if all values meet the criteria, false if not.---**

**[1,2,3,4,5,7].every(x=>x>0) //true**

**some ---Use the some Method to Check that Any Elements in an Array Meet a Criteria---**

**[1,2,3,4,5,7].some(x=>x>5) //true**

The some method works with arrays to check if any element passes a particular test. It returns a **Boolean value - true** if any of the values meet the criteria, false if not.

**---We'll pass you an array of two numbers. Return the sum of those two numbers plus the sum of all the numbers between them.---**

**function sumAll(arr = [5, 10]) {**

**Array.from({ length: Math.max(...arr) }, (a, b) => b + 1).splice(Math.min(...arr) - 1).reduce((x, y) => x + y)**

**}**

**function sumAll(arr = [5, 10]) {**

**[...Array(Math.max(...arr)).keys()].map(a => a + 1).splice(Math.min(...arr) - 1).reduce((a, b) => a + b)**

**}**

**symmetric difference ---Compare two arrays and return a new array with any items only found in one of the two given arrays, but not both. In other words, return the symmetric difference of the two arrays.---**

**function diffArray(arr1, arr2) {**

**return [**

**...diff(arr1, arr2),**

**...diff(arr2, arr1)**

**];**

**function diff(a, b) {**

**return a.filter(x => !b.includes(x))**

**}**

**}**

**--- Write a function that takes two or more arrays and returns a new array of unique values in the order of the original provided arrays.---**

**[...(new Set([].concat(...arr1)))]**

**1. Foreach:- Foreach takes a callback function and run that callback function on each element of array one by one.**

**2. Filter:- filter executes the callback and check its return value. If the value is true element remains in the resulting array but if the return value is false the element will be removed for the resulting array.**

Also take notice filter does not update the existing array it will return a new filtered array every time.

**3. Map:- it generate a new array based on your existing array.**

**4. Reduce:- reduce the array to one single value.**

**---JavaScript Hoisting---**

A variable can be used before it has been declared.

Hoisting is JavaScript's default behavior of moving all declarations to the top of the current scope

**Variables and constants declared with let or const are not hoisted!**