Basic JavaScript: Manipulate Arrays With push()

.push()takes one or more *parameters* and "pushes" them onto the end of the array.

var arr = [1,2,3];  
arr.push(4);  
// arr is now [1,2,3,4]

## Basic JavaScript: Manipulate Arrays With pop()

.pop()is used to "pop" a value off of the end of an array. We can store this "popped off" value by assigning it to a variable. In other words, .pop()removes the last element from an array and returns that element.

var myArray = [["John", 23], ["cat", 2]];

var removedFromMyArray = myArray.pop();

console.log(removedFromMyArray);

## Basic JavaScript: Manipulate Arrays With shift()

pop()always removes the last element of an array. What if you want to remove the first?

That's where .shift()comes in. It works just like .pop(), except it removes the first element instead of the last.

var myArray = [["John", 23], ["dog", 3]];

var removedFromMyArray = myArray.shift();

console.log(removedFromMyArray);

## Basic JavaScript: Manipulate Arrays With unshift()

## .unshift() works exactly like .push(), but instead of adding the element at the end of the array, unshift() adds the element at the beginning of the array.

var myArray = [["John", 23], ["dog", 3]];

myArray.shift();

myArray.unshift(["Paul",35]);