## **Javascript String Functions In Action**

**indexOf(substr, [start])**

**It** Searches and (if found) returns the index number of the searched character or substring within the string. If not found, -1 is returned. “Start” is an optional argument specifying the position within string to begin the search. Default is 0.

For Example

//indexOf(char/substring)

var sentence="Hi, my name is Sam!"

if (sentence.indexOf("Sam")!=-1)

alert("Sam is in there!")

**lastIndexOf(substr, [start])**

**It** Searches and (if found) returns the index number of the searched character or substring within the string. Searches the string from end to beginning. If not found, -1 is returned. “Start” is an optional argument specifying the position within string to begin the search. Default is string.length-1.

For Example

//lastIndexOf(substr, [start])

var myString = 'javascript rox';

console.log(myString.lastIndexOf('r'));

//output: 11

**charAt(x)**

It Returns the character at the “x” position within the string.

For Example

//charAt(x)

var myString = 'jQuery FTW!!!';

console.log(myString.charAt(7));

//output: F

**charCodeAt(x)**

It Returns the Unicode value of the character at position “x” within the string.

For Example

//charAt(position)

var message="jquery4u"

//alerts "q"

alert(message.charAt(1))

**concat(v1, v2,…)**

**It** Combines one or more strings (arguments v1, v2 etc) into the existing one and returns the combined string. Original string is not modified.

For Example

//concat(v1, v2,..)

var message="Sam"

var final=message.concat(" is a"," hopeless romantic.")

//alerts "Sam is a hopeless romantic."

alert(final)

**fromCharCode(c1, c2,…)**

**It** Returns a string created by using the specified sequence of Unicode values (arguments c1, c2 etc). Method of String object, not String instance. For example: String.fromCharCode().

//fromCharCode(c1, c2,...)

console.log(String.fromCharCode(97,98,99,120,121,122))

//output: abcxyz

console.log(String.fromCharCode(72,69,76,76,79))

//output: HELLO

## **JavaScript Array Functions In Action**

**toString()**

The JavaScript method toString() converts an array to a string of (comma separated) array values.

For Example

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
document.getElementById("demo").innerHTML = fruits.toString();

### **Result**

Banana,Orange,Apple,Mango

**join()**

The **join()** method also joins all array elements into a string.

It behaves just like toString(), but in addition you can specify the separator:

For Example

var fruits = ["Banana", "Orange","Apple", "Mango"];  
document.getElementById("demo").innerHTML = fruits.join(" \* ");

### **Result**

Banana \* Orange \* Apple \* Mango

## **Popping**

The pop() method removes the last element from an array:

For Example

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.pop();              // Removes the last element ("Mango") from fruits

## **Pushing**

The push() method adds a new element to an array (at the end):

For Example

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.push("Kiwi");       //  Adds a new element ("Kiwi") to fruits

## **Shifting Elements**

Shifting is equivalent to popping, working on the first element instead of the last.

The shift() method removes the first array element and "shifts" all other elements to a lower index.C

For Example

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.shift();            // Removes the first element "Banana" from fruits

The unshift() method adds a new element to an array (at the beginning), and "unshifts" older elements:

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.unshift("Lemon");    // Adds a new element "Lemon" to fruits