**Javascript fundamental**

\_Arrays, strings, objects

1. Convert from Arrays to Strings:

a).toString()

ex: let fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.toString();

// "Banana,Orange,Apple,Mango"

b) If I want to specify separators:

ex: var fruits = ["Banana", "Orange","Apple", "Mango"];

fruits.join(“ \* “)

// Banana \* Orange \* Apple \* Mango

1. Array.pop(): Pops out the last item of the array, changes the array

Ex:

var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.pop();

// fruits now doesn’t contain Mango

But let x = fruits.pop() => will give you “Mango”

1. Push: Add an item to the array, modify the array. RETURNs new array length

Ex:

var fruits = ["Banana", "Orange", "Apple", "Mango"];

var x = fruits.push("Kiwi"); // the value of x is 5

fruits now has 5 items

1. Shift(): removes the first array item, and shifts all other elements to a lower index

Ex: var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.shift(); // Removes the first element "Banana" from fruits

The shift() method returns the string that was "shifted out":

Example

var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.shift(); // Returns "Banana"

1. Unshift(): add a new element to an array at the beginning and "unshifts" older elements:

Example

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

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_array_unshift)

The unshift() method returns the new array length.

Example

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.unshift("Lemon");    // Returns 5

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_array_unshift_return)

1. Changing Elements, using index number:

var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits[0] = "Kiwi"; // Changes the first element of fruits to "Kiwi"

The length property provides an easy way to append a new element to an array:

Example

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits[fruits.length] = "Kiwi";          // Appends "Kiwi" to fruit

1. Splicing an array: Add new items to an array, and to remove them:

**ADD:**

var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.splice(2, 0, "Lemon", "Kiwi");

The first parameter (2) defines the position where new elements should be added (spliced in).

The second parameter (0) defines how many elements should be removed.

The rest of the parameters ("Lemon" , "Kiwi") define the new elements to be added.

**REMOVE:**

var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.splice(0, 1); // Removes the first element of fruits, remove 1 item starting 0

CONCAT Arrays;  
The **concat()** method creates a new array by merging (concatenating) existing arrays:

var myGirls = ["Cecilie", "Lone"];

var myBoys = ["Emil", "Tobias","Linus"];

var myChildren = myGirls.concat(myBoys); // Concatenates (joins) myGirls and myBoys

Example (Merging Three Arrays)

var arr1 = ["Cecilie", "Lone"];

var arr2 = ["Emil", "Tobias","Linus"];

var arr3 = ["Robin", "Morgan"];

var myChildren = arr1.concat(arr2, arr3); // Concatenates arr1 with arr2 and arr3

1. SLICE:

The slice() method slices out a piece of an array into a new array. Kind of like Splice, but with different syntaxes. Here, I can only slice out 1 single part of the array:

var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];  
var citrus = fruits.slice(1);

//Remove “Orange”

Can also take 2 arguments:

var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];  
var citrus = fruits.slice(1, 3);

NOT MODIFY THE ARRAY

Q: SPLICE vs SLICE?

A: Same thing, but SPLICE modifies the original array, cutting out the items that need to be cut out permanently

SLICE doesn’t modify the original array

Q: Sorting?

A: .sort() only sorts the array alphabetically

Ex: var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.sort();

.reverse() reverses the element in the array

Q: Numeric sort?

A:

1. Ascending order

var points = [40, 100, 1, 5, 25, 10];  
points.sort(function(a, b){return a - b});

1. Descending order:

var points = [40, 100, 1, 5, 25, 10];

points.sort(function(a, b){return b - a});

1. Sorting random?

Math.random()

Q: Find highest or lowest array value?

A:\_ Highest: Math.max.appl(null,arr)

Ex:

function myArrayMax(arr) {

return Math.max.apply(null, arr);

}

\_Lowest: Math.min.apply.bind(null, arr)

STRING:

Q: to get the index of the first occurrence of a specified text:

var str = "Please locate where 'locate' occurs!";  
var pos = str.indexOf("locate");

A: //return 7

Q:to get the index of the last occurrence of a specified text:

var str = "Please locate where 'locate' occurs!";  
var pos = str.lastIndexOf("locate");

A:// return 21

* They will return -1 if text not found

Both methods accept a second parameter as the starting position for the search:

Example

var str = "Please locate where 'locate' occurs!";  
var pos = str.indexOf("locate",15);

//First indexOf === str.search(“string”)

// But search() can take much more powerful search values (regular expressions), but cannot taje a second start position argument

Q: What are the methods to extract a part of a string?

A:

1. Slice( start, end)

slice() extracts a part of a string and returns the extracted part in a new string.

The method takes 2 parameters: the starting index (position), and the ending index (position).

This example slices out a portion of a string from position 7 to position 13:

Example

var str = "Apple, Banana, Kiwi";

var res = str.slice(7, 13);

The result of res will be:

Banana

1. If a parameter is negative, the position is counted from the end of the string.

This example slices out a portion of a string from position -12 to position -6:

Example

var str = "Apple, Banana, Kiwi";  
var res = str.slice(-12, -6);

The result of res will be:

Banana

If you omit the second parameter, the method will slice out the rest of the string:

Example

var res = str.slice(7);

## **The substring() Method**

**substring()** is similar to slice().

The difference is that substring() cannot accept negative indexes.

## **The substr() Method**

**substr()** is similar to slice().

The difference is that the second parameter specifies the **length** of the extracted part.

### Example

var str = "Apple, Banana, Kiwi";  
var res = str.substr(7, 6);

The result of res will be:

Banana

## **Replacing String Content**

The **replace()** method replaces a specified value with another value in a string:

### Example

str = "Please visit Microsoft!";  
var n = str.replace("Microsoft", "W3Schools");

By default, the replace() function replaces **only the first** match:

To replace all matches, use a **regular expression** with a **/g** flag (global match):

Example

str = "Please visit Microsoft and Microsoft!";  
var n = str.replace(/Microsoft/g, "W3Schools");

//Notice it’s /string/

To replace case insensitive, use a **regular expression** with an **/i** flag (insensitive):

Example

str = "Please visit Microsoft!";  
var n = str.replace(/MICROSOFT/i, "W3Schools");

A string is converted to upper case with **toUpperCase()**:

Example

var text1 = "Hello World!";       // String  
var text2 = text1.toUpperCase();  // text2 is text1 converted to upper

A string is converted to lower case with **toLowerCase()**:

Example

var text1 = "Hello World!";       // String  
var text2 = text1.toLowerCase();  // text2 is text1 converted to lower

The **concat()** method can be used instead of the plus operator. These two lines do the same:

Example

var text = "Hello" + " " + "World!";  
var text = "Hello".concat(" ", "World!");

EXTRACT STRING CHARACTERS:

Q: The **charAt()** method returns the character at a specified index (position) in a string:

Example

var str = "HELLO WORLD";  
str.charAt(0);            // returns H

**CAN CONVERT STRING TO ARRAY:**

A string can be converted to an array with the **split()** method:

Example

var txt = "a,b,c,d,e";   // String  
var x = txt.split(",");          // Split on commas  
var y = txt.split(" ");          // Split on spaces

// They return arrays

// x = ["a", "b", "c", "d", "e"]

// y = ["a,b,c,d,e"]

var txt = "Hello"; // String

txt.split(""); // Split in characters

**charCodeAt():** returns the first character in a string using Unicode:

ex:

var str = "HELLO WORLD";  
var n = str.charCodeAt(0);

The result of n will be:

72

SUMMARY

Q: splice/slice(starting, length starting from the starting point)

A: ex:

var fruits = ["Banana", "Orange", "Apple", "Kiwi"];

fruits.splice(1,2)

//Banana, Kiwi

Q: Concat string?

A:

var str1 = "Hello ";

var str2 = "World!";

str1.concat(" ",str2)

"Hello World!"

Q: Apply jQuery?

A:

<script>

var str1 = "Hello ";

var str2 = "World!";

document.getElementById("demo").innerHTML = str1.concat(str2);

</script>

<https://www.linkedin.com/learning/four-semesters-of-computer-science-in-5-hours/hash-table>

Q: indexOf?

A: var str= “Hello world”

var n = str.indexOf(“world”)

// will return 7