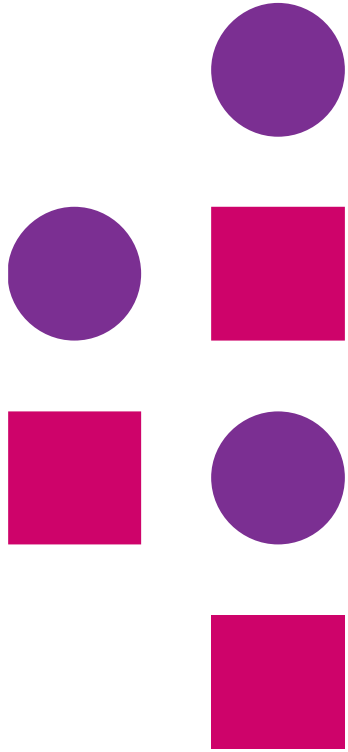
WSA | Forward looking IT finishing school

Strings and Array

JavaScript

Table of Content

- Strings
- Strings access methods
- Array
- Array access methods
- Multi dimensional arrays



Strings

(Introduction to Strings)

What is String?

- String is a sequence of Unicode characters.
- Unicode characters are encoding standard that has widespread acceptance.
- Some of the most commonly used character encodings are

Encoding	Description
UTF - 8	Widely used in email systems and on the internet.
UTF - 16	Used in windows, Java and by JavaScript, and often for plain text and for word-processing data files on Windows.
UTF - 32	The mainly used in internal APIs where the data is single code points or glyphs, rather than strings of characters.

Strings Syntax

```
<script>
    var str = " ";
    var str1 = "Hello World";
    var str2 = new String("Hi");
</script>
```

Strings - Immutable

- In JavaScript strings are immutable or unchangeable.
- An immutable object is an object whose state cannot be modified after it is created.

Example:

```
<script>
    var str = "Hello World";
    str[0] = "h";
    console.log(str); //will be "Hello World"
</script>
```

String Methods

(Pre-defined methods to perform various operations in strings)

Strings Methods

Method	Description
length	Returns the length of the string
concat()	Joins the string with one or more strings
trim()	Removes the white spaces from beginning and end of the string
replace()	Replaces old string with the new string
substring()	Returns the part of string
substr()	Returns the part of string

Strings Methods

Method	Description
indexOf()	Returns the position of specified word
lastIndexOf()	Returns the last occurrence of specified word
slice()	Returns the part of the string
split()	Converts string into array
includes()	Returns true if string includes the specified word.

The length() Method

- The length method returns the size of the string.

```
Syntax :
var length = string.length;
```

```
Example :
  var str = "Hello World";
  var len = str.length;
  document.write("Length of the string " + len);
```

The concat() Method

- The concat() method will join the string with one or more strings.

```
Syntax :

var str1 = str.concat("hello"," ","World");

                    or

var str2 = str.concat(str1,str2,str3…..strN);
```

The concat() Method

```
Example :

var str1 = "Hello World";
var str2 = "Welcome";
var concatString = str1.concat(" ", str2, " ", "to WSA");
document.write("Concatenation of string" + concatString);
```

The trim() Method

- The trim() method will remove the spaces from beginning and end of the strings.

```
Syntax :
var result = string.trim();
```

```
Example :

var str1 = "   Hello World   ";
var trimmedString = str1.trim();
console.log ("After trim" +trimmedString);
```

The trimStart() or trimLeft() Method

- The trimStart() or trimLeft() methods will remove the spaces from the beginning of the strings.

```
Syntax :
var result = string.trimLeft();
             Or
var result = string.trimStart();
```

The trimStart() or trimLeft() Method

```
Example :

var str1 = "    Hello World    ";
var trimmedStart = str1.trimStart();
var trimmedLeft  =  str1.trimLeft();
console.log("After trim" +trimmedStart);
console.log("<br>After trim" +trimmedLeft);
```

The trimEnd() or trimRight() Method

- The trimEnd() or trimRight() methods will remove the spaces from the beginning of the strings.

```
Syntax :
var result = string.trimEnd();
             Or
var result = string.trimRight();
```

The trimEnd() or trimRight() Method

```
Example :
var str1 = "   Hello World   ";
var trimmedEnd = str1.trimEnd();
var trimmedRight  =  str1.trimRight();
console.log("After trim" +trimmedEnd);
console.log("<br>After trim" +trimmedRight);
```

The replace() Method

- The replace() method will replace the specified string into new string.

```
Syntax :

var replacedStr = str.replace('string to replace','new string');

Or

var replacedStr = str.replace(RegularExpression, 'new string');
```

The replace() Method

```
Example 1:

var str1 =“Welcome to JS World”;
var repStr = str1.replace(‘JS’,’JavaScript’);
console.log(“Replaced String” +repStr);
```

```
Example 2:
var str =“A blue bottle with a blur liquid is on a blue table Blue”;
var replacedStr = str.replace(/blue/g, ”green”);
document.write(“After replace” +replacedStr);
```

The substring() Method

- The substring() method will return the part of the strings.

```
Syntax :

var subStr = str.substring(startIndex, EndIndex);
```

```
Example :

 var str1 =“Javascript World”;
 var subStr = str1.substring(0,10);
 document.write(“Subtring is: ” +subStr);
```

The substr() Method

- The substr() method will return the part of the strings.

```
Syntax :

var subStr = str.substr(startIndex, numberOfCharacters);
```

```
Example :

 var str1 ="Javascript World";
 var subStr = str1.substr(0,10);
 document.write("Subtring is: " +subStr);
```

The indexOf() Method

The indexOf() method will return the position of first occurrence of the specified string

```
Syntax :

var index = str.indexOf("string");
```

```
Example :

var str1 ="Javascript World";
var indexStr = str1.indexOf("World");
document.write("Index of world" +indexStr);
```

The lastIndexOf() Method

The lastIndexOf() method will return the position of last occurrence of the specified string

```
Syntax :

var lastIndex = str.lastIndexOf("string");
```

```
Example :

var str1 ="Javascript World! Welcome to World";
var lastIndex = str1.lastIndexOf("World");
document.write("Last Index of world" +lastIndex);
```

Exercise

Write a JavaScript program to extract the user name and domain name.

- Accept email address from the user, extract the user name and the domain name, for
- example,

Input: abc.xyz@gmail.com

Output: username: abc.xyz

domain : gmail.com

The slice() Method

- The slice() method will return the part of the strings.

```
Syntax :

var slicedStr = str.slice(startIndex, EndIndex);
```

```
Example :

var str ="Javascript world";
var slicedStr = str.slice(1,5);
document.write("Sliced String" +slicedStr);
```

The split() Method

- The split() method will convert string into array of strings.

```
Syntax :

var splitStr = str.split(separator,limit);
```

```
Example :

var str ="Javascript world. Welcome to WSA";
var splitStr = str.split(" ")
document.write("Array of string" +splitStr);
var splitLimit = str.split(" ",2);
document.write("Array of string" +splitLimit);
```

The includes() Method

The includes() method will true if string includes the specified word else returns the false.

Syntax :

```
var includesStr = str.includes(searchString, position);
```

Example :

```
var str = 'To be, or not to be, that is the question.';
console.log(str.includes('To be'));
console.log(str.includes('question',0));
console.log(str.includes('nonexistent'));
```

Exercise

Convert the given string into title case with following rules:

- Capitalize the first letter of each word. Capitalize nouns, pronouns, adjectives, verbs, adverbs, and subordinate conjunctions.
- Lowercase articles (a, an, the), coordinating conjunctions, and prepositions (under, between, over, on, at).
- Lowercase the 'to' in an infinitive (I want to play guitar).

Examples:

- How Are You Doing Today?
- Our Office between Metro and Barton Centre
- What Is the Need of This Expensive Phone?

Exercise

1. Count number of palindromes in a given string:

2. Special Palindrome

- In English we call a particular string as palindrome if it reads the same way from left and right (ex: Malayalam, mom etc..)
- However there are some special kind of palindromes are there which will have:
  1. Upper case characters
  2. Special characters
  3. Spaces in-between.

Examples:

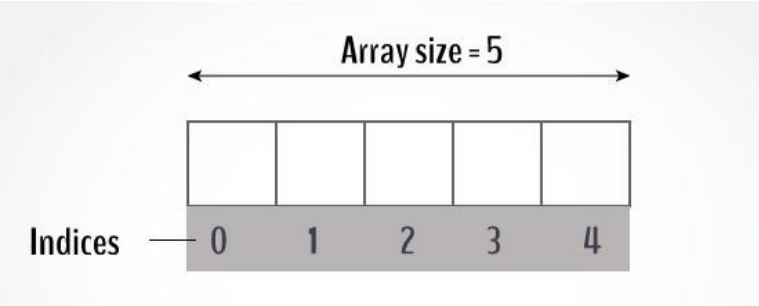
1. "Madam, I'm Adam"
2. "A nut for a jar of tuna."
3. "Are we not pure? "No, sir!" Panama's moody Noriega brags. It is garbage!" Irony dooms a man—a prisoner up to new era."

Arrays

(Introduction to Arrays)

What is an Array?

- Array is a collection of similar or different data types
- Each data in array is called an element
- Each elements has a numeric position, known as its index / indices, in the array
- The index numbers start from zero
- In JavaScript, Array is also an object. The `typeof` operator will return the same
- Array object has `length` property which returns the total number of elements



Array
(Syntax – Using [ ])

```
Syntax :
// Creates initialized array
var array-name = [item1, item2, …];

// Creates empty
var array-name = [ ];
```

Array
(Syntax – Using constructor)

```
Syntax :
var array-name = new Array(item1, item2, item3);
Or
var array-name = Array(item1, item2, item3);
Or
var array-name = Array(array-length);
```

Array - Example

```
Example :
var array = [10, 20, 30, 40, 50];
Or
var array = new Array(10, 20, 30, 40, 50);
Or
var array = Array(3);
```

Array Access – Using loop

```
for ( let idx = 0; idx < 5; idx++ ) {
    document.write("Element at index " + idx + " is " + myArray[idx]);
}
```

```
for ( let idx in myArray ) {
    document.write("Element at index " + idx + " is " + myArray[idx]);
}
```

Array - Heterogeneous

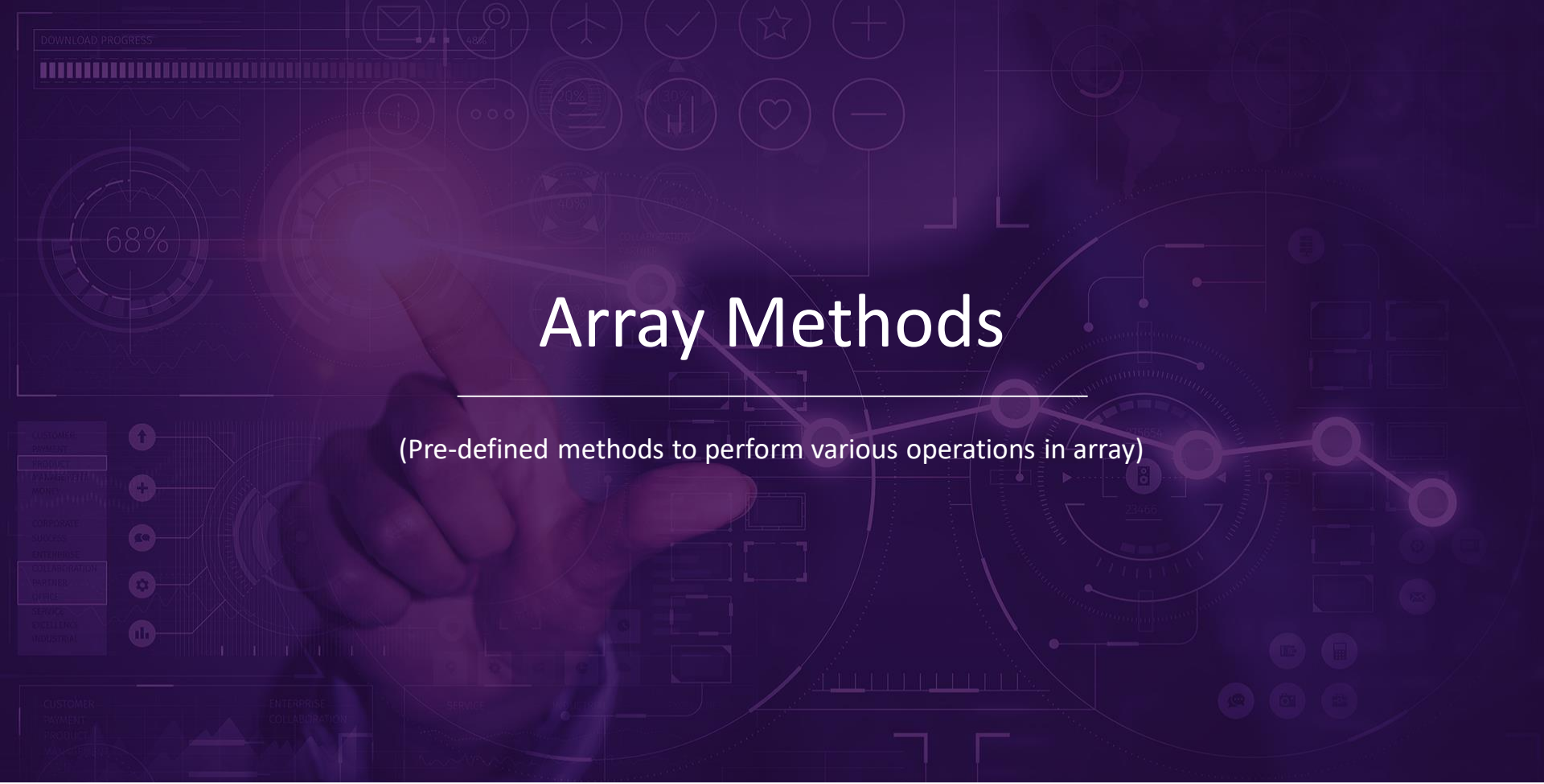
Each element in an array can be of different types

```
var mixArray = [“Student”, 95, true];

for(let idx = 0; idx < 3; idx++) {
    console.log (mixArray[idx]);
}
```

Exercise

- WAP to create an array with 10 integers and find out the following:
  - Sum of all number
  - Average
  - Maximum value & Minimum value
- WAP to reverse an array elements.



Array Methods

(Pre-defined methods to perform various operations in array)

Array Methods

Method	Description
`join()`	Concatenates all the elements of array into string
`pop()`	Deletes last element of an array
`push()`	Appends new element in the last of array
`sort()`	Sorts an array in alphabetical order
`reverse()`	Reverses array elements order in the array

Array Methods

Method	Description
`shift()`	Removes first element from the array and shifts all other element to a lower index
`unshift()`	Unshift method adds elements to the beginning of an array and return new length of array
`slice()`	The array slice method returns part of an array
`splice()`	The array splice method can be used for adding and/or removing elements from an array

Array Methods

Method	Description
`Filter()`	Creates a new array with all elements that pass the test implemented by the provided function.
`map()`	Creates a new array with the results of calling a provided function on each element in the calling array.

The join() Method

- The join() method converts all the elements of an array to strings and concatenates them, returning the resulting string
- It behaves like toString(), but we can specify separator

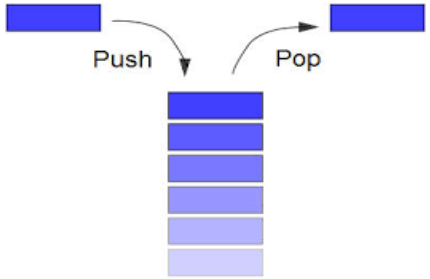
```
Example :
var fruits = ["Banana", "Orange","Apple", "Mango"];
var str = fruits.join(" + ");
document.write(str);
```

The push() and pop() methods

Stack operations – LIFO

```
var numStack = [10, 20, 30, 40, 50];
numStack.pop();  // Pop an element
for (let idx=0; idx <numStack.length; idx++) {
    document.write(numStack[idx] + "<br>");
}

numStack.push(100); // Push an element
for (let idx=0; idx <numStack.length; idx++) {
    document.write(numStack[idx] + "<br>");
}
```



The shift() methods

The **shift()** method removes the **first** element from an array and returns that removed element.

```
Syntax
array.shift();
```

```
Example:
var array1 = [1, 2, 3];
var firstElement = array1.shift();
console.log(array1);// expected output: Array [2, 3]
console.log(firstElement);// expected output: 1
```

The unshift() methods

The **unshift()** method adds one or more elements to the beginning of an array and returns the new length of the array.

```
Syntax
array.unshift();
```

```
Example:
var array1 = [1, 2, 3];
array1.unshift(2);
console.log(array1);
array1.unshift(5,8);
console.log(array1);
```

The sort() Method

The sort() method sorts an array in alphabetic order

```
var numList = [‘banana’, ‘apple’, ‘mango’, ‘grapes’];
numList.sort();
for (let idx = 0; idx < numList.length; idx++){
    document.write(numList[idx] + "<br>");
}
```

The reverse() Method

The reverse() method reverses the elements in an array

```
var numList = [55, 3, 16, 21];
numList.reverse();
for(let idx = 0; idx < numList.length; idx++){
    document.write(numList[idx] + "<br>");
}
```

Exercise

- WAP to represent two sets of integers. Find out union and intersection of those two sets (Ref – Set theory)
  - Assume there are duplicates in the array
  - Hint – Write a findElement() function to check if an element is present in an array or not
- WAP to find Nth largest element in a given array
- WAP to perform shift operations of a given element
  - Shift Nth element by right by M positions
  - Shift Nth element by left by M positions

The splice() Method

The splice() method will change the content of original array by removing or adding the element.

```
Syntax:

array.splice(startIndex, number of elements, elements to add);
```

The splice() Method

```
var ar = [1, 4, 9, 10, 11, 16];

ar.splice(1,3);// will remove the elements

console.log(ar);

ar.splice(1,0,5);// will insert the element 5 at index 1

console.log(ar);
```

The forEach() Method

The map() method creates a new array with the results of calling a provided function on each element in the calling array.

```
Syntax:

arrayName.forEach(callback_function(currentItem, index, array) {

      //Application Logic

});
```

The forEach() Method

```
Example:

var age = [21,19,34,56,23,20,17,65,76,15,35,14,13,25];

age.forEach(function(item) {
    document.write(item+" ");
});
```

The filter() Method

Filter() method creates a new array with all elements that pass the test implemented by the provided function.

```
Syntax:

var new_array =
current_array.filter(function(currentItem,index,array)
{
    //Application Logic to filter current array items
        //return items for new array
});
```

The filter() Method

Example:

```
var words = ['limit', 'elite', 'exuberant', 'destruction‘];

function word(elem) {
    return elem.length > 6;
}
var result = words.filter(word);
console.log(result);
// expected output: Array ["exuberant", "destruction‟];
```

Exercise

Write a JavaScript program to squeeze a word in given sentence.

For example:

1. Input : what is what so what
   Output: what is so

2. Input : I mean they can't be so mean
   Output : I mean they can't be so

3. Input : Butler bought bitter butter but he made it as a better butter by adding sweeter butter
   Output : Butler bought bitter butter but he made it as a better by adding sweeter

The map() Method

The map() method creates a new array with the results of calling a provided function on each element in the calling array.

Syntax:

```
var new_array = current_array.map(function(currentItem, index, array) {
      //Application Logic
      //Return elements for new array
});
```

The map() Method

Example:

```
var curArray = [1, 4, 9, 16];// pass a function to map

var doubleArray = curArray.map(doubleItem);

function doubleItem(currentItem) {
    return 2 * currentItem;
}
console.log(doubleArray);
```

Exercise

Write a JavaScript program to greet the guest.

For example:

1. Input : [‘Ram’,’Radhika’,’John’,’Sumit’]
   Output: Welcome Ram
   Welcome Radhika
   Welcome John
   Welcome Sumit

Multidimensional Array

The matrix

Multidimensional Array

Declaring Multidimensional Array

```
var array2d = [ [10, 20, 30], [40, 50, 60], [70, 80, 90] ];
for(let idx = 0; idx < array2d.length; idx++) {
  for(let jdx = 0; jdx < array2d[idx].length; jdx++) {
    document.write(array2d[idx][jdx] + " ");
  }
  document.write("<br>");
}
```

Exercise

- WAP to find out sum of diagonal elements in a 2D array
- WAP to find max and min value in a given row
- WAP to multiply two matrix using 2D arrays

WebStack Academy

#83, Farah Towers,
1st Floor, MG Road,
Bangalore – 560001

M: +91-809 555 7332
E: training@webstackacademy.com

WSA in Social Media: