JAVASCRIPT - THE ARRAYS OBJECT

The **Array** object lets you store multiple values in a single variable. It stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Syntax

Use the following syntax to create an **Array** object −

```
var fruits = new Array( "apple", "orange", "mango" );
```

The **Array** parameter is a list of strings or integers. When you specify a single numeric parameter with the Array constructor, you specify the initial length of the array. The maximum length allowed for an array is 4,294,967,295.

You can create array by simply assigning values as follows −

```
var fruits = [ "apple", "orange", "mango" ];
```

You will use ordinal numbers to access and to set values inside an array as follows.

```
fruits[0] is the first element
fruits[1] is the second element
fruits[2] is the third element
```

Array Properties

Here is a list of the properties of the Array object along with their description.

Property	**Description**
constructor	Returns a reference to the array function that created the object.
index	The property represents the zero-based index of the match in the string
input	This property is only present in arrays created by regular expression matches.
length	Reflects the number of elements in an array.
prototype	The prototype property allows you to add properties and methods to an object.

In the following sections, we will have a few examples to illustrate the usage of Array properties.

Array Methods

Here is a list of the methods of the Array object along with their description.

Method	**Description**
	Returns a new array comprised of this array joined with other arrays and/or

concat	value$s$.
every	Returns true if every element in this array satisfies the provided testing function.
filter	Creates a new array with all of the elements of this array for which the provided filtering function returns true.
forEach	Calls a function for each element in the array.
indexOf	Returns the first $least$ index of an element within the array equal to the specified value, or -1 if none is found.
join	Joins all elements of an array into a string.
lastIndexOf	Returns the last $greatest$ index of an element within the array equal to the specified value, or -1 if none is found.
map	Creates a new array with the results of calling a provided function on every element in this array.
pop	Removes the last element from an array and returns that element.
push	Adds one or more elements to the end of an array and returns the new length of the array.
reduce	Apply a function simultaneously against two values of the array $fromleft - to - right$ as to reduce it to a single value.
reduceRight	Apply a function simultaneously against two values of the array $fromright - to - left$ as to reduce it to a single value.
reverse	Reverses the order of the elements of an array -- the first becomes the last, and the last becomes the first.
shift	Removes the first element from an array and returns that element.
slice	Extracts a section of an array and returns a new array.
some	Returns true if at least one element in this array satisfies the provided testing function.
toSource	Represents the source code of an object
sort	Represents the source code of an object
	Adds and/or removes elements from an array.

splice	
toString	Returns a string representing the array and its elements.
unshift	Adds one or more elements to the front of an array and returns the new length of the array.

In the following sections, we will have a few examples to demonstrate the usage of Array methods.