JavaScript Objects, Properties, and Methods

1. Document Object

The document object represents the whole html document.

- When html document is loaded in the browser, it becomes a document object. It is the root element that represents the html document.

Methods

We can access and change the contents of document by its methods.

- The important methods of document object are as follows:

Method	Description
write("string")	writes the given string on the doucment.
writeln("string")	writes the given string on the doucment with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

Accessing the field value by document object

In this example, we are going to get the value of input text by user. Here, we are using document.form1.name.value to get the value of name field.

Here, document is the root element that represents the html document.

form1 is the name of the form.

name is the attribute name of the input text.

value is the property, that returns the value of the input text.

Let's see the simple example of document object that prints name with welcome message.

```
<script type="text/javascript">
function printvalue(){
var name=document.form1.name.value;
alert("Welcome: "+name);
}
</script>

<form name="form1">
Enter Name:<input type="text" name="name"/>
<input type="button" onclick="printvalue()" value="print name"/>
</form>
```

Properties

Property	Description
alinkcolor	The color to be used when displaying activated links in the document
anchors	An array of anchor objects present in the document
bgColor	The background color of the document
fgColor	The foreground color (i.e. the color of the text in the document)
forms	An array of the Forms objects contained in the document.
images	An Array of Image objects contained the document
lastModified	The date that the document was last modified
linkColor	The color to be used when displaying links in the document
links	An array of links contained in the document
vlinkColor	The color to be used when displaying links in the document which have been visited by the current user
innerHTML	The innerHTML property can be used to modify your document's HTML.When you use innerHTML, you can change the page's content without refreshing the page.

Writing Text to a document in a different Window

The writing of text to a document is not restricted to writing to the document in the current window. It is also possible to write content to a different window simply by referencing the window in the JavaScript command (for details on opening new windows see the JavaScript Window Object chapter). The following example opens a new window and then writes text into that new window:

```
<html>
<head>
<title>JavaScript Document Object Example</title>
<script language="javascript" type="text/javascript">

function writeContent(content, windowObj)
{
      windowObj.document.write(content);
}
</script>
</head>

<body>

<script language="javascript" type="text/javascript">

newWindowObj = window.open ("", "MyWindow");

</script>

<form action="">
     <input type=button value="Write to New Window"
onclick="writeContent('This is new content in the new window',
newWindowObj)"/>
</form>
```

```
</body>
</html>
```

Changing the Document Title

Even when the title of a document has been specified using the HTML <title> tag it is possible to dynamically change the current setting using the title property of the document object.

The following example initially sets the document title to JavaScript Document Object Example and provides a button which, when pressed, changes the title to a new value:

```
<html>
<head>
<title>JavaScript Document Object Example</title>
</head>
<body>
<form action="">
    <input type=button value="Press to change title"
onclick="document.title='hello'"/>
</form>
</body>
</html>
```

Changing the Document Colors

A similar approach can be used to change the colors using in the document.

The following example changes the foreground and background colors when the "Change colors" button is pressed:

```
<html>
<head>
```

```
<title>JavaScript Document Object Example</title>
<script language="javascript" type="text/javascript">

function changeColors()
{
	document.fgColor="red";
	document.bgColor="blue";
}
</script>
</head>

<body>
<p>
Sample text to show color change
</p>
<form action="">
     <input type=button value="Change colors"
onclick="changeColors()"/>
</form>
</body>
</html>
```

Getting a List of Objects in a Document

the links property of the document object can be used to obtain a list of Link objects (in other words, links to other web pages) in the current document.

To demonstrate this, the following example consists of a simple HTML page containing two links to other sites (Amazon.com and Yahoo). The J

```
<html>
<head>
<title>JavaScript Document Object Example</title>
```

```
</head>

<body>

<a href="http://www.amazon.com">Buy more books</a>
<br>
<a href="http://www.yahoo.com">Stop making Google rich</a>
<br>

<script language="javascript" type="text/javascript">
for (i in document.links)
{
	document.write( document.links[i] + "<br>" );
}
</script>

</form>
</body>
</html>
```

JavaScript Array Object

- Describes the JavaScript array object including parameters, properties, and methods.

Parameters

- arrayLength
- elementN - Array element list of values

Properties

- index
- input
- length - The quantity of elements in the object.

Methods

- **concat()** -Join two arrays:

```
<script>
function myFunction() {
   var hege = ["Cecilie", "Lone"];
   var stale = ["Emil", "Tobias", "Linus"];
   var children = hege.concat(stale);
   document.write(children);
}
</script>
```

- **join(delimiter)** - Puts all elements in the array into a string, separating each element with the specified delimiter.

```
Var words = new
Array("limit","lines","finish","complete","In","Out")
var jwords = words.join(";")
```

The value of the string jwords is:

limit;lines;finish;complete;In;Out

- **pop()** - Pops the last string off the array and returns it.

```
Var words = new
Array("limit","lines","finish","complete","In","Out")

var lastword = words.pop()
```

The value of the string lastword is:

Out

- **push(strings)** - Strings are placed at the end of the array.

```
Var words = new Array("limit","lines","finish")
var a=words.push("complete","In","Out")

document.write(a);
```

The array, words, will be:

limit, lines, finish, complete, In, Out

- **reverse()** - Puts array elements in reverse order.

```
Var words = new
Array("limit","lines","finish","complete","In","Out")
var a=words.reverse()

    document.write(a);
```

The array, words, will be:

Out, In, complete, finish, lines, limit

- **shift()** - Decreases array element size by one by shifting the first element off the array and returning it.

```
Var words = new
Array("limit","lines","finish","complete","In","Out")
Var w = words.shift()

   document.write(w);
```

The array, w, will be:

In, complete, finish, lines, limit

The string word will be:

Out

- **sort()** - Sorts the array elements in dictionary order

  var arr = new Array("orange", "mango", "banana", "sugar");

var sorted = arr.sort();

document.write("Returned string is : " + sorted);

The value of words becomes:

banana,mango,orange,sugar

- **splice()** - It is used to take elements out of an array and replace them with those specified. In the below example the element starting at element 3 is removed, two of them are removed and replaced with the specified strings. The value returned are those values that are replaced.

  Var words = new Array("limit","lines","finish","complete","In","Out")
  Var words1 = words.splice(3, 2, "done", "On")

  document.write(words1);

  The value of words1 becomes:

  limit, lines, finish, done, On, Out

  The value of words1 is set to:

  complete, In

- **split(deliimiter)** - Splits a string using the delimiter and returns an array.

  Var words = new String("limit;lines;finish;complete;In;Out")
  var swords = words.split(";")

```
document.write(swords);
```

The values in the array swords is:

```
limit, lines, finish, complete, In, Out
```

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

```
 <script type="text/javascript">
  var arr = new Array("orange", "mango", "banana", "sugar");

  var length = arr.unshift("water");
  document.write("Returned array is : " + arr );
  document.write("<br /> Length of the array is : " + length );
</script>
```

Output

```
Returned array is : water,orange,mango,banana,sugar
Length of the array is : 5
```

JavaScript - The Strings Object

➢ The String object lets you work with a series of characters; it wraps Javascript's string primitive data type with a number of helper methods.

- As JavaScript automatically converts between string primitives and String objects, you can call any of the helper methods of the String object on a string primitive.

Syntax

Use the following syntax to create a String object −

```
var val = new String(string);
```

The String parameter is a series of characters that has been properly encoded.

String Properties

Here is a list of the properties of String object and their description.

Property	**Description**
length	Returns the length of the string.

String Methods

Here is a list of the methods available in String object along with their description.

Method	**Description**	**Example**

charAt()	Returns the character at the specified index.	var str = new String("This is string");
		document.writeln("str.charAt(0) is:" + str.charAt(0));
		document.writeln("<br/>str.charAt(1) is:" + str.charAt(1));
		document.writeln("<br/>str.charAt(2) is:" + str.charAt(2));
		Output
		str.charAt(0) is:T
		str.charAt(1) is:h
		str.charAt(2) is:i

concat()	Combines the text of two strings and returns a new string.	<script type="text/javascript">
		var str1 = new String("This is string one");
		var str2 = new String("This is string two");
		var str3 = str1.concat(str2);
		
		document.write("Concatenated String :" + str3);
		</script>
		
		Output
		Concatenated String :This is string oneThis is string two.

indexOf()	Returns the index within the calling String object	<script type="text/javascript"> var str1 = new String("This is string one"); var index = str1.indexOf("string"); document.write("indexOf found String :"

	of the first occurrence of the specified value, or -1 if not found.	+ index); document.write("<br />"); var index = str1.indexOf("one"); document.write("indexOf found String :" + index); </script> **Output** indexOf found String :8 indexOf found String :15
lastIndexOf()	Returns the index within the calling String object of the last occurrence of the specified value, or -1 if not found.	<script type="text/javascript"> var str1 = new String("This is string one and again string"); var index = str1.lastIndexOf("string"); document.write("lastIndexOf found String :" + index); document.write("<br />"); var index = str1.lastIndexOf("one"); document.write("lastIndexOf found String :" + index); </script> **Output** lastIndexOf found String :29 lastIndexOf found String :15
replace()	Used to find	myString = new String ("This is my string");

	a match between a regular expression and a string, and to replace the matched substring with a new substring.	var newString = myString.replace ("my", "your"); Output 'This is your string'

slice()	Extracts a section of a string and returns a new string.	<script>
		var s1="abcdefgh";
		var s2=s1.slice(2,5);
		document.write(s2);
		</script>
		</body>
		output
		cde

split()	Splits a String object into an array of strings by separating the string	<script type="text/javascript">
		var str = "Apples are round, and apples are juicy.";
		var splitted = str.split(" ", 3);
		
		document.write(splitted);

	into substrings.	`</script>` **Output** Apples,are,round,

substr()	Returns the characters in a string beginning at the specified location through the specified number of characters.	`<script type="text/javascript">`
		`var str = "Apples are round, and apples are juicy.";`
		
		`document.write("(1,2): " + str.substr(1,2));`
		`document.write("<br />(1): " + str.substr(1));`
		`document.write("<br />(20, 2): " + str.substr(20,2));`
		`</script>`
		
		`</body>`
		`</html>`
		Output
		(1,2): pp
		(1): pples are round, and apples are juicy.
		
		(20, 2): d

substring()	Returns the characters in a string between two indexes into	`myString = new String ("This is my string");`
		`var partString = myString.substring(5, 10);`
		
		output
		
		'is my'

	the string.	
toLowerCase()	Returns the calling string value converted to lower case.	<script> var s1="JavaScript toLowerCase Example"; var s2=s1.toLowerCase(); document.write(s2); </script>
toString()	Returns a string representing the specified object.	<script type="text/javascript"> var str = "Apples are round, and Apples are Juicy."; document.write(str.toString( )); </script> Output Apples are round, and Apples are Juicy.
toUpperCase()	Returns the calling string value converted to uppercase.	<script> var s1="JavaScript toUpperCase Example"; var s2=s1.toUpperCase(); document.write(s2); </script>
Trim()	This method	<script>

removes leading and trailing whitespaces from the string.	var s1=" javascript trim ";
	var s2=s1.trim();
	document.write(s2);
	</script>

Date Object

The JavaScript date object can be used to get year, month and day. You can display a timer on the webpage by the help of JavaScript date object.

You can use different Date constructors to create date object. It provides methods to get and set day, month, year, hour, minute and seconds.

Constructor

You can use 4 variant of Date constructor to create date object.

1. Date()
2. Date(milliseconds)
3. Date(dateString)
4. Date(year, month, day, hours, minutes, seconds, milliseconds)

Here is a description of the parameters −

- **No Argument** − With no arguments, the Date() constructor creates a Date object set to the current date and time.
- **milliseconds** − When one numeric argument is passed, it is taken as the internal numeric representation of the date in milliseconds, as returned by the getTime() method. For example, passing the argument 5000 creates a date that represents five seconds past midnight on 1/1/70.

- **datestring** − When one string argument is passed, it is a string representation of a date, in the format accepted by the Date.parse() method.(Javascript date.parse() method takes a date string and returns the number of milliseconds since midnight of January 1, 1970.)

  Example

```
<script type="text/javascript">
       var msecs = Date.parse( "Aug 28, 2008 23:30:00" );
       document.write( "Number of milliseconds from 1970: " +
msecs );
     </script>

   </body>
</html>
```

Output

```
Number of milliseconds from 1970: 1219946400000
```

- 7 agruments − To use the last form of the constructor shown above. Here is a description of each argument:
  - **year** − Integer value representing the year., you should always specify the year in full; use 1998, rather than 98.
  - **month** − Integer value representing the month, beginning with 0 for January to 11 for December.
  - **date** − Integer value representing the day of the month.
  - **hour** − Integer value representing the hour of the day (24-hour scale).
  - **minute** − Integer value representing the minute segment of a time reading.
  - **second** − Integer value representing the second segment of a time reading.
  - **millisecond** − Integer value representing the millisecond segment of a time reading.

JavaScript Date Methods

The important methods of date object are as follows:

Method	Description
getFullYear()	returns the year in 4 digit e.g. 2015. It is a new method and suggested than getYear() which is now deprecated.
getMonth()	returns the month in 2 digit from 1 to 31.
getDate()	returns the date in 1 or 2 digit from 1 to 31.
getDay()	returns the day of week in 1 digit from 0 to 6.
getHours()	returns all the elements having the given name value.
getMinutes()	returns all the elements having the given class name.
getSeconds()	returns all the elements having the given class name.
getMilliseconds()	returns all the elements having the given tag name.

Math Object

Methods

The following table lists the methods available with the Math object:

Method	Description	Example

abs	Absolute value	`<script type="text/javascript">`
		`  var value = Math.abs(-1);`
		`        document.write("First Test Value : " + value );`
		
		` var value = Math.abs(null);`
		`        document.write("<br />Second Test Value : " + value );`
		
		`        var value = Math.abs(20);`
		`        document.write("<br />Third Test Value : " + value );`
		
		`        var value = Math.abs("string");`
		`        document.write("<br />Fourth Test Value : " + value );`
		`      </script>`
		
		Output
		First Test Value : 1
		Second Test Value : 0
		Third Test Value : 20
		Fourth Test Value : NaN

sin, cos, tan	Standard trigonometric functions; argument in radians	<script type="text/javascript"> var value =Math.cos(90); document.write("First Test Value : " + value); var value =Math.cos(30); document.write("<br />Second Test Value : " + value); var value = Math.sin(90); document.write("<br />Second Test Value : " + value); var value = Math.tan(45); document.write("<br />Third Test Value : " + value); </script>
exp, log	Exponential and natural logarithm, base e	<script type="text/javascript"> var value = Math.exp(1); document.write("First Test Value : " + value); var value = Math.log(10); document.write("First Test Value : " + value); </script>
ceil	Returns least integer greater than or equal to argument	<body> Ceil of 4.6 is:

		<script> document.write(Math.ceil(4.6)); </script> Output Ceil of 4.6 is: 5
floor	Returns greatest integer less than or equal to argument	<body> Floor of 4.6 is: <script> document.write(Math.floor(4.6)); </script> </body> Output Floor of 4.6 is: 4

min, max	Returns greater or lesser (respectively) of two arguments	var value = Math.min(10, 20, -1, 100);
		document.write("First Test Value : " + value);
		
		var value = Math.min(-1, -3, -40);
		document.write("<br />Second Test Value : " + value);

		Output
		First Test Value : -1
		Second Test Value : -40
		
		<script type="text/javascript">
		        var value = Math.max(10, 20, -1, 100);
		        document.write("First Test Value : " + value);
		
		        var value = Math.max(-1, -3, -40);
		        document.write("<br />Second Test Value : " + value);
		Output
		First Test Value : 100
		Second Test Value : -1

pow	Exponential; first argument is base, second is exponent	<body> 3 to the power of 4 is: <script> document.write(Math.pow(3,4)); </script> </body> Output

		3 to the power of 4 is: 81

random	Returns a random number between 0 and 1.	`<body>`
		
		Random Number is:
		
		`<script>`
		
		`document.write(Math.random());`
		
		`</script>`
		
		`</body>`

round	Rounds argument to nearest integer	`<script type="text/javascript">`
		
		`var value = Math.round( 20.7 );`
		`document.write("<br />Second Test Value : " + value );`
		
		`var value = Math.round( 20.3 );`
		`document.write("<br />Third Test Value : " + value );`
		
		`var value = Math.round( -20.3 );`
		`document.write("<br />Fourth Test Value : " + value );`
		`</script>`
		
		`</body>`
		`</html>`
		
		Output
		Second Test Value : 21

		Third Test Value : 20 Fourth Test Value : -20
sqrt	Square root	<body> Square Root of 17 is: <script> document.write(Math.sqrt(17)); </script> </body> Output Square Root of 17 is: 4.123105625617661

Window Object
The window object represents a window in browser. An object of window is created automatically by the browser.

Methods

Method	Description
alert()	Displays the alert box containing message with ok button.
confirm()	Displays the confirm dialog box containing message with ok and cancel button.

prompt()	Displays a dialog box to get input from the user.
open()	Opens the new window.
close()	Closes the current window.
setTimeout()	It calls a function or evaluates an expression after a specified number of milliseconds.