**EXPERIMENT NO: 3**

**a. Write a JavaScript code for implementing Date object, Math object, Array object, Document object and Event object.**

**Theory**

# Date Object

The Date object is a datatype built into the JavaScript language. Date objects are created with the **new Date( )** as shown below.

Once a Date object is created, a number of methods allow you to operate on it. Most methods simply allow you to get and set the year, month, day, hour, minute, second, and millisecond fields of the object, using either local time or UTC (universal, or GMT) time.

The ECMAScript standard requires the Date object to be able to represent any date and time, to millisecond precision, within 100 million days before or after 1/1/1970. This is a range of plus or minus 273,785 years, so the JavaScript is able to represent date and time till year 275755.

# Syntax:

Here are different variant of Date() constructor:

|  |
| --- |
| new Date( )new Date(milliseconds)new Date(datestring)new Date(year,month,date[,hour,minute,second,millisecond ]) |

**Note:** Paramters in the brackets are always optional

Here is the 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.
* **7 agruments:** To use the last form of constructor given above, Here is the description of each argument:
	1. **year:** Integer value representing the year. For compatibility (in order to avoid the Y2K problem), you should always specify the year in full; use 1998, rather than 98.
	2. **month:** Integer value representing the month, beginning with 0 for January to 11 for December.
	3. **date:** Integer value representing the day of the month.
	4. **hour:** Integer value representing the hour of the day (24-hour scale).
	5. **minute:** Integer value representing the minute segment of a time reading.
	6. **second:** Integer value representing the second segment of a time reading.
	7. **millisecond:** Integer value representing the millisecond segment of a time reading.

# Date Properties:

Here is a list of each property and their description.

|  |  |
| --- | --- |
| **Property** | **Description** |
| Constructor | Specifies the function that creates an object's prototype. |
| Prototype | The prototype property allows you to add properties and methods to an object. |

# Date Methods:

Here is a list of each method and its description.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [Date()](http://www.tutorialspoint.com/javascript/date_date.htm) | Returns today's date and time |
| [getDate()](http://www.tutorialspoint.com/javascript/date_getdate.htm) | Returns the day of the month for the specified date according to local time. |
| [getDay()](http://www.tutorialspoint.com/javascript/date_getday.htm) | Returns the day of the week for the specified date according to local time. |
| [getFullYear()](http://www.tutorialspoint.com/javascript/date_getfullyear.htm) | Returns the year of the specified date according to local time. |
| [getHours()](http://www.tutorialspoint.com/javascript/date_gethours.htm) | Returns the hour in the specified date according to local time. |
| [getMilliseconds()](http://www.tutorialspoint.com/javascript/date_getmilliseconds.htm) | Returns the milliseconds in the specified date according to local time. |
| [getMinutes()](http://www.tutorialspoint.com/javascript/date_getminutes.htm) | Returns the minutes in the specified date according to local time. |
| [getMonth()](http://www.tutorialspoint.com/javascript/date_getmonth.htm) | Returns the month in the specified date according to local time. |
| [getSeconds()](http://www.tutorialspoint.com/javascript/date_getseconds.htm) | Returns the seconds in the specified date according to local time. |
| [getTime()](http://www.tutorialspoint.com/javascript/date_gettime.htm) | Returns the numeric value of the specified date as the number of milliseconds since January 1, 1970, 00:00:00 UTC. |
| [getTimezoneOffset()](http://www.tutorialspoint.com/javascript/date_gettimezoneoffset.htm) | Returns the time-zone offset in minutes for the current locale. |
| [getUTCDate()](http://www.tutorialspoint.com/javascript/date_getutcdate.htm) | Returns the day (date) of the month in the specified date according to universal time. |
| [getUTCDay()](http://www.tutorialspoint.com/javascript/date_getutcday.htm) | Returns the day of the week in the specified date according to universal time. |
| [getUTCFullYear()](http://www.tutorialspoint.com/javascript/date_getutcfullyear.htm) | Returns the year in the specified date according to universal time. |
| [getUTCHours()](http://www.tutorialspoint.com/javascript/date_getutchours.htm) | Returns the hours in the specified date according to universal time. |
| [getUTCMilliseconds()](http://www.tutorialspoint.com/javascript/date_getutcmilliseconds.htm) | Returns the milliseconds in the specified date according to universal time. |
| [getUTCMinutes()](http://www.tutorialspoint.com/javascript/date_getutcminutes.htm) | Returns the minutes in the specified date according to universal time. |
| [getUTCMonth()](http://www.tutorialspoint.com/javascript/date_getutcmonth.htm) | Returns the month in the specified date according to universal time. |
| [getUTCSeconds()](http://www.tutorialspoint.com/javascript/date_getutcseconds.htm) | Returns the seconds in the specified date according to universal time. |
| [getYear()](http://www.tutorialspoint.com/javascript/date_getyear.htm) | **Deprecated** - Returns the year in the specified date according to local time. Use getFullYear instead. |
| [setDate()](http://www.tutorialspoint.com/javascript/date_setdate.htm) | Sets the day of the month for a specified date according to local time. |
| [setFullYear()](http://www.tutorialspoint.com/javascript/date_setfullyear.htm) | Sets the full year for a specified date according to local time. |
| [setHours()](http://www.tutorialspoint.com/javascript/date_sethours.htm) | Sets the hours for a specified date according to local time. |
| [setMilliseconds()](http://www.tutorialspoint.com/javascript/date_setmilliseconds.htm) | Sets the milliseconds for a specified date according to local time. |
| [setMinutes()](http://www.tutorialspoint.com/javascript/date_setminutes.htm) | Sets the minutes for a specified date according to local time. |
| [setMonth()](http://www.tutorialspoint.com/javascript/date_setmonth.htm) | Sets the month for a specified date according to local time. |
| [setSeconds()](http://www.tutorialspoint.com/javascript/date_setseconds.htm) | Sets the seconds for a specified date according to local time. |
| [setTime()](http://www.tutorialspoint.com/javascript/date_settime.htm) | Sets the Date object to the time represented by a number of milliseconds since January 1, 1970, 00:00:00 UTC. |
| [setUTCDate()](http://www.tutorialspoint.com/javascript/date_setutcdate.htm) | Sets the day of the month for a specified date according to universal time. |
| [setUTCFullYear()](http://www.tutorialspoint.com/javascript/date_setutcfullyear.htm) | Sets the full year for a specified date according to universal time. |
| [setUTCHours()](http://www.tutorialspoint.com/javascript/date_setutchours.htm) | Sets the hour for a specified date according to universal time. |
| [setUTCMilliseconds()](http://www.tutorialspoint.com/javascript/date_setutcmilliseconds.htm) | Sets the milliseconds for a specified date according to universal time. |
| [setUTCMinutes()](http://www.tutorialspoint.com/javascript/date_setutcminutes.htm) | Sets the minutes for a specified date according to universal time. |
| [setUTCMonth()](http://www.tutorialspoint.com/javascript/date_setutcmonth.htm) | Sets the month for a specified date according to universal time. |
| [setUTCSeconds()](http://www.tutorialspoint.com/javascript/date_setutcseconds.htm) | Sets the seconds for a specified date according to universal time. |
| [setYear()](http://www.tutorialspoint.com/javascript/date_setyear.htm) | **Deprecated -** Sets the year for a specified date according to local time. Use setFullYear instead. |
| [toDateString()](http://www.tutorialspoint.com/javascript/date_todatestring.htm) | Returns the "date" portion of the Date as a human-readable string. |
| [toGMTString()](http://www.tutorialspoint.com/javascript/date_togmtstring.htm) | **Deprecated -** Converts a date to a string, using the Internet GMT conventions. Use toUTCString instead. |
| [toLocaleDateString()](http://www.tutorialspoint.com/javascript/date_tolocaledatestring.htm) | Returns the "date" portion of the Date as a string, using the current locale's conventions. |
| [toLocaleFormat()](http://www.tutorialspoint.com/javascript/date_tolocaleformat.htm) | Converts a date to a string, using a format string. |
| [toLocaleString()](http://www.tutorialspoint.com/javascript/date_tolocalestring.htm) | Converts a date to a string, using the current locale's conventions. |
| [toLocaleTimeString()](http://www.tutorialspoint.com/javascript/date_tolocaletimestring.htm) | Returns the "time" portion of the Date as a string, using the current locale's conventions. |
| [toSource()](http://www.tutorialspoint.com/javascript/date_tosource.htm) | Returns a string representing the source for an equivalent Date object; you can use this value to create a new object.  |
| [toString()](http://www.tutorialspoint.com/javascript/date_tostring.htm) | Returns a string representing the specified Date object. |
| [toTimeString()](http://www.tutorialspoint.com/javascript/date_totimestring.htm) | Returns the "time" portion of the Date as a human-readable string. |
| [toUTCString()](http://www.tutorialspoint.com/javascript/date_toutcstring.htm) | Converts a date to a string, using the universal time convention. |
| [valueOf()](http://www.tutorialspoint.com/javascript/date_valueof.htm) | Returns the primitive value of a Date object.  |

# Date Static Methods:

In addition to the many instance methods listed previously, the Date object also defines two static methods. These methods are invoked through the Date( ) constructor itself:

|  |  |
| --- | --- |
| **Method** | **Description** |
| [Date.parse( )](http://www.tutorialspoint.com/javascript/date_parse.htm) | Parses a string representation of a date and time and returns the internal millisecond representation of that date. |
| [Date.UTC( )](http://www.tutorialspoint.com/javascript/date_utc.htm) | Returns the millisecond representation of the specified UTC date and time. |

**Example on Date Object**

<html>

<body>

<script type="text/javascript">

var d = new Date()

document.write(d.getDate())

document.write(".")

document.write(d.getMonth() + 1)

document.write(".")

document.write(d.getFullYear())

</script>

</body>

</html>

**Output**



**Math Object**

The **math** object provides you properties and methods for mathematical constants and functions.

Unlike the other global objects, *Math* is not a constructor. All properties and methods of Math are static and can be called by using *Math* as an object without creating it.

Thus, you refer to the constant pi as **Math.PI** and you call the *sine* function as **Math.sin(x)**, where x is the method's argument.

# Syntax:

Here is the simple syntax to call properties and methods of Math.

|  |
| --- |
| var pi\_val = Math.PI;var sine\_val = Math.sin(30); |

# Math Properties:

Here is a list of each property and their description.

|  |  |
| --- | --- |
| **Property** | **Description** |
| [E](http://www.tutorialspoint.com/javascript/math_e.htm)  | Euler's constant and the base of natural logarithms, approximately 2.718. |
| [LN2](http://www.tutorialspoint.com/javascript/math_ln2.htm)  | Natural logarithm of 2, approximately 0.693. |
| [LN10](http://www.tutorialspoint.com/javascript/math_ln10.htm)  | Natural logarithm of 10, approximately 2.302. |
| [LOG2E](http://www.tutorialspoint.com/javascript/math_log2e.htm)  | Base 2 logarithm of E, approximately 1.442. |
| [LOG10E](http://www.tutorialspoint.com/javascript/math_log10e.htm)  | Base 10 logarithm of E, approximately 0.434. |
| [PI](http://www.tutorialspoint.com/javascript/math_pi.htm)  | Ratio of the circumference of a circle to its diameter, approximately 3.14159. |
| [SQRT1\_2](http://www.tutorialspoint.com/javascript/math_sqrt1_2.htm)  | Square root of 1/2; equivalently, 1 over the square root of 2, approximately 0.707. |
| [SQRT2](http://www.tutorialspoint.com/javascript/math_sqrt2.htm)  | Square root of 2, approximately 1.414. |

# Math Methods

Here is a list of each method and its description.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [abs()](http://www.tutorialspoint.com/javascript/math_abs.htm) | Returns the absolute value of a number. |
| [acos()](http://www.tutorialspoint.com/javascript/math_acos.htm) | Returns the arccosine (in radians) of a number. |
| [asin()](http://www.tutorialspoint.com/javascript/math_asin.htm) | Returns the arcsine (in radians) of a number. |
| [atan()](http://www.tutorialspoint.com/javascript/math_atan.htm) | Returns the arctangent (in radians) of a number. |
| [atan2()](http://www.tutorialspoint.com/javascript/math_atan2.htm) | Returns the arctangent of the quotient of its arguments. |
| [ceil()](http://www.tutorialspoint.com/javascript/math_ceil.htm) | Returns the smallest integer greater than or equal to a number. |
| [cos()](http://www.tutorialspoint.com/javascript/math_cos.htm) | Returns the cosine of a number. |
| [exp()](http://www.tutorialspoint.com/javascript/math_exp.htm) | Returns EN, where N is the argument, and E is Euler's constant, the base of the natural logarithm. |
| [floor()](http://www.tutorialspoint.com/javascript/math_floor.htm) | Returns the largest integer less than or equal to a number. |
| [log()](http://www.tutorialspoint.com/javascript/math_log.htm) | Returns the natural logarithm (base E) of a number. |
| [max()](http://www.tutorialspoint.com/javascript/math_max.htm) | Returns the largest of zero or more numbers. |
| [min()](http://www.tutorialspoint.com/javascript/math_min.htm) | Returns the smallest of zero or more numbers. |
| [pow()](http://www.tutorialspoint.com/javascript/math_pow.htm) | Returns base to the exponent power, that is, base exponent. |
| [random()](http://www.tutorialspoint.com/javascript/math_random.htm) | Returns a pseudo-random number between 0 and 1. |
| [round()](http://www.tutorialspoint.com/javascript/math_round.htm) | Returns the value of a number rounded to the nearest integer. |
| [sin()](http://www.tutorialspoint.com/javascript/math_sin.htm) | Returns the sine of a number. |
| [sqrt()](http://www.tutorialspoint.com/javascript/math_sqrt.htm) | Returns the square root of a number. |
| [tan()](http://www.tutorialspoint.com/javascript/math_tan.htm) | Returns the tangent of a number. |
| [toSource()](http://www.tutorialspoint.com/javascript/math_tosource.htm) | Returns the string "Math". |

**Example on Math Object**

<html>

<body>

<script type="text/javascript">

document.write("Round of 7.25 : "+Math.round(7.25));

document.write("<br>");

document.write("Random : " +Math.random());

document.write("<br>");

document.write("Max : "+Math.max(2,4));

document.write("<br>");

document.write("Min : "+Math.min(2,4));

document.write("<br>");

var no=Math.random()\*10;

document.write("Floor : " +Math.floor(no));

document.write("<br>");

document.write("Euler's number : "+Math.E);

document.write("<br>");

document.write("PI : "+Math.PI);

document.write("<br>");

document.write("square root of 2 : "+Math.SQRT2);

document.write("<br>");

document.write("square root of 1/2 : "+Math.SQRT1\_2);

document.write("<br>");

document.write("natural logarithm of 2 : "+Math.LN2);

document.write("<br>");

document.write("natural logarithm of 10 : "+Math.LN10);

document.write("<br>");

document.write("base 2 logarithm of E : "+Math.LOG2E)

document.write("<br>");

document.write("base 10 logarithm of E : "+Math.LOG10E)

</script>

</body>

</html>

**Output :**

****

**Array Object**

The **Array** object let's you store multiple values in a single variable.

# Syntax:

Creating a **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 each property and their description.

|  |  |
| --- | --- |
| **Property** | **Description** |
| [constructor](http://www.tutorialspoint.com/javascript/array_constructor.htm)  | 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](http://www.tutorialspoint.com/javascript/array_length.htm)  | Reflects the number of elements in an array. |
| [prototype](http://www.tutorialspoint.com/javascript/object_prototype.htm) | The prototype property allows you to add properties and methods to an object. |

# Array Methods

Here is a list of each method and its description.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [concat()](http://www.tutorialspoint.com/javascript/array_concat.htm) | Returns a new array comprised of this array joined with other array(s) and/or value(s). |
| [every()](http://www.tutorialspoint.com/javascript/array_every.htm) | Returns true if every element in this array satisfies the provided testing function. |
| [filter()](http://www.tutorialspoint.com/javascript/array_filter.htm) | Creates a new array with all of the elements of this array for which the provided filtering function returns true. |
| [forEach()](http://www.tutorialspoint.com/javascript/array_foreach.htm) | Calls a function for each element in the array. |
| [indexOf()](http://www.tutorialspoint.com/javascript/array_indexof.htm) | Returns the first (least) index of an element within the array equal to the specified value, or -1 if none is found. |
| [join()](http://www.tutorialspoint.com/javascript/array_join.htm) | Joins all elements of an array into a string. |
| [lastIndexOf()](http://www.tutorialspoint.com/javascript/array_lastindexof.htm) | Returns the last (greatest) index of an element within the array equal to the specified value, or -1 if none is found. |
| [map()](http://www.tutorialspoint.com/javascript/array_map.htm) | Creates a new array with the results of calling a provided function on every element in this array. |
| [pop()](http://www.tutorialspoint.com/javascript/array_pop.htm) | Removes the last element from an array and returns that element. |
| [push()](http://www.tutorialspoint.com/javascript/array_push.htm) | Adds one or more elements to the end of an array and returns the new length of the array. |
| [reduce()](http://www.tutorialspoint.com/javascript/array_reduce.htm) | Apply a function simultaneously against two values of the array (from left-to-right) as to reduce it to a single value. |
| [reduceRight()](http://www.tutorialspoint.com/javascript/array_reduceright.htm) | Apply a function simultaneously against two values of the array (from right-to-left) as to reduce it to a single value. |
| [reverse()](http://www.tutorialspoint.com/javascript/array_reverse.htm) | Reverses the order of the elements of an array -- the first becomes the last, and the last becomes the first. |
| [shift()](http://www.tutorialspoint.com/javascript/array_shift.htm) | Removes the first element from an array and returns that element. |
| [slice()](http://www.tutorialspoint.com/javascript/array_slice.htm) | Extracts a section of an array and returns a new array. |
| [some()](http://www.tutorialspoint.com/javascript/array_some.htm) | Returns true if at least one element in this array satisfies the provided testing function. |
| [toSource()](http://www.tutorialspoint.com/javascript/array_tosource.htm) | Represents the source code of an object |
| [sort()](http://www.tutorialspoint.com/javascript/array_sort.htm) | Sorts the elements of an array. |
| [splice()](http://www.tutorialspoint.com/javascript/array_splice.htm) | Adds and/or removes elements from an array. |
| [toString()](http://www.tutorialspoint.com/javascript/array_tostring.htm) | Returns a string representing the array and its elements.  |
| [unshift()](http://www.tutorialspoint.com/javascript/array_unshift.htm) | Adds one or more elements to the front of an array and returns the new length of the array. |

**Example of Array**

<html>

<head>

<title>JavaScript Array length Property</title>

</head>

<body>

<script type="text/javascript">

 var arr = new Array( "one", "two", "three");

 document.write("arr.length is : " + arr.length);

 document.write("<br>");

 document.write("Value at 1 st location : " + arr[0]);

 document.write("<br>");

 document.write("Value at 2 nd location : " + arr[1]);

 document.write("<br>");

 document.write("Value at 3 rd location : " + arr[2]);

</script>

</body>

</html>

**Output**

****

# JavaScript - Document Object Model or DOM

Every web page resides inside a browser window which can be considered as an object.

A Document object represents the HTML document that is displayed in that window. The Document object has various properties that refer to other objects which allow access to and modification of document content.

The way that document content is accessed and modified is called the **Document Object Model**, or **DOM**. The Objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a Web document.

* **Window object:** Top of the hierarchy. It is the outmost element of the object hierarchy.
* **Document object:** Each HTML document that gets loaded into a window becomes a document object. The document contains the content of the page.
* **Form object:** Everything enclosed in the <form>...</form> tags sets the form object.
* **Form control elements:** The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

Here is a simple hierarchy of few important objects:



There are several DOMs in existence. The following sections explain each of these DOMs in detail and describe how you can use them to access and modify document content.

* [The Legacy DOM:](http://www.tutorialspoint.com/javascript/javascript_legacy_dom.htm) This is the model which was introduced in early versions of JavaScript language. It is well supported by all browsers, but allows access only to certain key portions of documents, such as forms, form elements, and images.
* [The W3C DOM:](http://www.tutorialspoint.com/javascript/javascript_w3c_dom.htm) This document object model allows access and modification of all document content and is standardized by the World Wide Web Consortium (W3C). This model is supported by almost all the modern browsers.
* [The IE4 DOM:](http://www.tutorialspoint.com/javascript/javascript_ie4_dom.htm) This document object model was introduced in Version 4 of Microsoft's Internet Explorer browser. IE 5 and later versions include support for most basic W3C DOM features.