[JavaScript math object](http://mrbool.com/math-functions-course-javascript-part-11/17914" \o "Math Functions - JavaScript Course - Part 11" \t "_blank) contains various built-in methods and mathematical constants for performing mathematical calculations in scripts. The Math object of JavaScript allows you to perform certain calculations by using method functions of the Math object. It provides a few constants such as pi, log values and exponential values. For this we have to define a variable and set its value through a property or function of the Math Object.

If you want to use a math property which returns a constant value, then you have to write like this, for getting value of pi in a simple variable you would have to use.

[ var value=math.property ]

[var pi\_value=Math.pi ] //This variable return value of pi 3.14

[var my\_money=Math.sqrt(x);]

Most of the member functions have one or more parameters, which is what the "x" is for. You can replace x with a number or variable. For instance, if you want the square root of a number, you can call the square root member function of the Math object with the number as the parameter.

[ var my\_money=Math.sqrt(16); ]

This gives back the square root of 4.

There are few rules and regulation for using math object; you have to remember while using math object. The "M" in Math is always capitalized. The word Math is followed by the dot operator (.), and then the property or member function you want to use. Here we will deal with some most common method and properties.

**Listing 1:** Script to finding round off of a number

<html>

<head>

<title>JavaScript Math Object</title>

</head>

<body>

<script type="text/javascript">

document.write(Math.round(16.12345) + "<br />")

document.write(Math.round(25.0)+ "<br />")

document.write(Math.round(36.111) + "<br />")

document.write(Math.round(49.992222233) + "<br />")

</script>

</body>

</html>

Above figure is output of round function used in Math object ,which round off the passed no

Above is the output of round function used in Math object ,which round off the passed no.

Lets discuss another example the max() method of the Math object which returns the largest of the given numbers given in arguments of the max method.

**Listing 2:** Script to find max number between two number

<html>

<head>

<title>JavaScript Math Object</title>

</head>

<body>

<script type="text/javascript">

document.write(Math.max(3000,2999, 4999) + "<br />")

document.write(Math.max(-2000,5000) + "<br />")

document.write(Math.max(-2,-7, 0, +0, 0.1) + "<br />")

document.write(Math.max(22,-7) + "<br />")

</script>

</body>

</html>

Above figure is output of math function

Above is output of math function

Another function used is the sqrt() function. Following listed script will compile the idea of sqrt() function.

**Listing 3:** Script to calculate squire root of given number

<html>

<body>

<script type="text/javascript">

document.write(Math.sqrt(16) + "<br />")

document.write(Math.sqrt(25) + "<br />")

document.write(Math.sqrt(36) + "<br />")

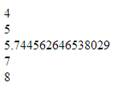
document.write(Math.sqrt(49) + "<br />")

document.write(Math.sqrt(64) + "<br />")

</script>

</body>

</html>



There is another example, Assume you have a point with the (x,y) coordinates of (14,18), you could calculate the angle in radians between that point and the positive X axis as follows. For this purpose Math object provide a function atan2(X, ).following example shows the idea of atan2(X,Y). Below calculating angle in both radian and degree. To convert radian into degree, multiply radian with 180\*7/22

**Listing 4:** Script to calculate angle

<html>

<head>

<title>JavaScript Math Object</title>

</head>

<body>

<script type="text/javascript">

document.write(Math.atan2(8,8)\*180/Math.PI + "<br />")

document.write(Math.atan2(32,16)\*180/Math.PI+ "<br />")

document.write(Math.atan2(64,128)\*180/Math.PI + "<br />")

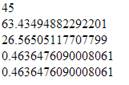
document.write(Math.atan2(128,256) + "<br />")

document.write(Math.atan2(256,512) + "<br />")

</script>

</body>

</html>



**Figure 4:**Above figure is output of atain2 () function in which atan2() method returns the arctangent of the quotient of its arguments, as a numeric value between PI and -PI radians. The number returned represents the counter clockwise angle in radians (not degrees) between the positive X axis and the point (x, y).

JavaScript provide the eight mathematical constant that can be handle by JavaScript Math objects these are: ln2 , ln10, log2e, log10e, e, pi, sqrt2, sqrt1\_2 . And provide various mathematical function which is handle by math object of JavaScript in various way.

JavaScript Math pow( ) Method

Below is the example of the **Math pow()** Method.

* **Example:**

|  |
| --- |
| <script type="text/javascript">      document.write(Math.pow(3, 4));  </script> |

* **Output:**

81

The **Math.pow()** method is used to power of a number i.e., the value of number raised to some exponent. Since the Math.pow() is a static method of Math and therefore it is always used as Math.pow() and not as a method of an object created of Math class.

**Syntax:**

Math.pow(base, exponent)

**Parameters:** This method accepts two parameters as mentioned above and described below:

* **base:**It is the base number which is to be raised.
* **exponent:**It is the value used to raise the **base**.

**Return Value:** The **Math.pow()** method returns a number representing the given **base** raised to the power of the given **exponent**.

More codes for the above method are as follows:

**Program 1:** When both **base** and **exponent** are passed as positive numbers in parameters:

|  |
| --- |
| <script type="text/javascript">      document.write(Math.pow(9, 3));  </script> |

**Output:**

729

**Program 2:** When the **base** value is negative and the **exponent** is positive:

|  |
| --- |
| <script type="text/javascript">      document.write(Math.pow(-9, 3));  </script> |

**Output:**

-729

**Program 3:** When the **base** value is positive and the **exponent** is negative:

|  |
| --- |
| <script type="text/javascript">      document.write(Math.pow(9, -3));  </script> |

**Output:**

0.0013717421124828531

**Program 4:** When the **base** value is negative and the **exponent** has decimal point:

|  |
| --- |
| <script type="text/javascript">      document.write(Math.pow(-9, 0.5));  </script> |

**Output:**

NaN

**JavaScript Math Object**

**Examples**

**The Math object**

The built-in Math object includes mathematical constants and functions. You do not need to create the Math object before using it.

To store a random number between 0 and 1 in a variable called "r\_number":

|  |
| --- |
| r\_number=Math.random() |

To store the rounded number of 8.6 in a variable called "r\_number":

|  |
| --- |
| r\_number=Math.round(8.6) |

**The Most Common Methods**

**NN**: Netscape, **IE**: Internet Explorer, **ECMA**: Web Standard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Methods** | **Explanation** |  |  |  |
| max(x,y) | Returns the number with the highest value of x and y |  |  |  |
| min(x,y) | Returns the number with the lowest value of x and y |  |  |  |
| random() | Returns a random number between 0 and 1 |  |  |  |
| round(x) | Rounds x to the nearest integer |  |  |  |

[Round](http://www.w3schools.com/js/tryit.asp?filename=tryjs_math_round)  
How to round a specified number to the nearest whole number

|  |
| --- |
| <html> <body> <script type="text/javascript"> document.write(Math.round(7.25)) </script> </body> </html> |

[Random number](http://www.w3schools.com/js/tryit.asp?filename=tryjs_math_random)  
The random method returns a random number between 0 and 1

|  |
| --- |
| <html> <body> <script type="text/javascript"> document.write(Math.random()) </script> </body> </html> |

[Random number from 0-10](http://www.w3schools.com/js/tryit.asp?filename=tryjs_math_random09)  
How to write a random number from 0 to 10, using the round and the random method.

|  |
| --- |
| <html> <body> <script type="text/javascript"> no=Math.random()\*10 document.write(Math.floor(no)) </script> </body> </html> |

[Max number](http://www.w3schools.com/js/tryit.asp?filename=tryjs_math_max)  
How to test which of two numbers, has the highest value.

|  |
| --- |
| <html> <body> <script type="text/javascript"> document.write(Math.max(2,4)) </script> </body> </html> |

[Min number](http://www.w3schools.com/js/tryit.asp?filename=tryjs_math_min)  
How to test which of two numbers, has the lowest value.

|  |
| --- |
| <html> <body> <script type="text/javascript"> document.write(Math.min(2,4)) </script> </body> </html> |

**JavaScript Date Object**

Here we cover the basics of date object and then explain the different methods and property of date object.

Description:

The Date object is useful when you want to display a date or use a timestamp on page. In JavaScript date object is inbuilt data type. Once you created the object you can perform different kind of methods and property on the date object.

You can get and set the year, month, day, hour, minute, second, and millisecond fields of the date object by using different method. Date object uses either co-ordinated universal time (Universal/GMT) or local time for displaying date information.

According to ECMAScript (Standard of scripting language ) Date object can represent any date and time, to millisecond precision, within 100 million days before or after 1/1/1970. I.e.This is a range of plus or minus 273,785 years, So the JavaScript can show the date and time till year 275755.

Basically Date object is used to work with dates and times. Date objects are created using “new Date ()” syntax. Date () object Accept these listed arguments.

[Var a=new date() ] - without any argument empty Date ( ) constructor creates a Date object set to the current date and time.

[ var a=new date(miliseconds) ]- After entering any numerical value date() object calculates it in millisecond value as returned by the getTime() method .

[ var a=new date(datestring) ] - When single string argument is passed, it is a string representation of a date.

[ var a=new date(year,month,date,hour,minute,second,millisecond) ] - Year: Integer value representing the year. For compatibility (in order to avoid the Y2K problem), you should always specify the year in full; use 1992, rather than 92. Month: month, beginning with 0 for January to 11 for December. Date: day of the month. Hour: hour of the day (24-hour scale). Minute: minute segment of a time reading. Second: second segment of a time reading. Millisecond: Integer value representing the millisecond segment of a time reading.

JavaScript - Current Date:

JavaScript creates a Date object based on user's system clock and this is best way let JavaScript simply use the system clock. During creation of date object based on system clock it is important to note that if someone’s clock is off by a few hours or they are in other GMT then date object will give other times from the one created on your computer.

Let’s take some example to understand the concept of date object. If you are not passing any argument to the date constructer then it will create date object based on user’s system clock.

**Listing 1:** Script for date object

<html>

<head>

<title> - JavaScript Dates</title>

<script type="text/javascript">

var vDate = new Date()

var vMonth = vDate.getMonth() + 1

var vDay = vDate.getDate()

var vYear = vDate.getFullYear()

document.write("Today's Date: " + vMonth + "/" + vDay + "/" + vYear);

document.write("</br>");

document.write("Today's Date: " + vDate);

</script>

</head>

<body >

</body>

</html>

Script to start date object

**Figure 1:** Above figure is output of Script to start date object.

These are the listed method which is supported by JavaScript .we can print more date information by using these method

* getFullYear() - The four digit year (1970-9999)
* getMonth() - Number of month (0-11)
* getDate() - Day of the month (0-31)
* getDay() - Day of the week(0-6). 0 = Sunday, ... , 6 = Saturday
* getHours() - Number of hours (0-23)
* getMinutes() - Number of minutes (0-59)
* getSeconds() - Number of seconds (0-59)
* getTime() - Number of milliseconds

JavaScript - Current Time:

Now instead of displaying the date, Let’s display the format like digital clock .

**Listing 2:** Script of digital clock

<html>

<head>

<title>JavaScript Current Time</title>

</head>

<body>

<b>Current Time:

<script type="text/javascript">

var objDate = new Date()

var objHours = objDate.getHours()

var objMinutes = objDate.getMinutes()

if (objMinutes < 10){

objMinutes = "0" + objMinutes

}

document.write(objHours + ":" + objMinutes + " ")

if(objHours > 11){

document.write("PM")

} else {

document.write("AM")

}

</script>

</b>

</body>

</html>

showing current time in digital clock format

**Figure 2:** Above figure is showing current time in digital clock format.

It is important to know that if hours or minutes variable is less than 10 then we need to add a zero to the beginning of minutes. This is not necessary, but if it is 3:08 PM then it will display output "3.8 PM".

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **JavaScript Date Object**  **The Date object**  The Date object is used to work with dates and times.  You create an instance of the Date object with the "new" keyword.  To store the current date in a variable called "my\_date":   |  | | --- | | var my\_date=new Date() |   After creating an instance of the Date object, you can access all the methods of the object from the "my\_date" variable. If, for example, you want to return the date (from 1-31) of a Date object, you should write the following:   |  | | --- | | my\_date.getDate() |   You can also write a date inside the parentheses of the Date() object, like this:   |  | | --- | | new Date("Month dd, yyyy hh:mm:ss")  new Date("Month dd, yyyy")  new Date(yy,mm,dd,hh,mm,ss)  new Date(yy,mm,dd)  new Date(milliseconds) |   Here is how you can create a Date object for each of the ways above:   |  | | --- | | var my\_date=new Date("October 12, 1988 13:14:00")  var my\_date=new Date("October 12, 1988")  var my\_date=new Date(88,09,12,13,14,00)  var my\_date=new Date(88,09,12)  var my\_date=new Date(500) |   **The Most Common Methods**  **NN**: Netscape, **IE**: Internet Explorer, **ECMA**: Web Standard   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Methods** | **Explanation** | **NN** | **IE** | **ECMA** | | Date() | Returns a Date object | 2.0 | 3.0 | 1.0 | | getDate() | Returns the date of a Date object (from 1-31) | 2.0 | 3.0 | 1.0 | | getDay() | Returns the day of a Date object (from 0-6. 0=Sunday, 1=Monday, etc.) | 2.0 | 3.0 | 1.0 | | getMonth() | Returns the month of a Date object (from 0-11. 0=January, 1=February, etc.) | 2.0 | 3.0 | 1.0 | | getFullYear() | Returns the year of the Date object (four digits) | 4.0 | 4.0 | 1.0 | | getHours() | Returns the hour of the Date object (from 0-23) | 2.0 | 3.0 | 1.0 | | getMinutes() | Returns the minute of the Date object (from 0-59) | 2.0 | 3.0 | 1.0 | | getSeconds() | Returns the second of the Date object (from 0-59) | 2.0 | 3.0 | 1.0 |   **Examples**  [Date](http://www.w3schools.com/js/tryit.asp?filename=tryjs_datedate) Returns today's date including date, month, and year. Note that the getMonth method returns 0 in January, 1 in February etc. So add 1 to the getMonth method to display the correct date.   |  | | --- | | <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> |   [Time](http://www.w3schools.com/js/tryit.asp?filename=tryjs_datetime) Returns the current local time including hour, minutes, and seconds. To return the GMT time use getUTCHours, getUTCMinutes etc.   |  | | --- | | <html> <body> <script type="text/javascript"> var d = new Date() document.write(d.getHours()) document.write(".") document.write(d.getMinutes() + 1) document.write(".") document.write(d.getSeconds()) </script> </body> </html> |   [Set date](http://www.w3schools.com/js/tryit.asp?filename=tryjs_datesetfullyear) You can also set the date or time into the date object, with the setDate, setHour etc. Note that in this example, only the FullYear is set.   |  | | --- | | <html> <body> <script type="text/javascript"> var d = new Date() d.setFullYear("1990") document.write(".") </script> </body> </html> |   [UTC time](http://www.w3schools.com/js/tryit.asp?filename=tryjs_dateutcdate) The getUTCDate method returns the Universal Coordinated Time which is the time set by the World Time Standard.   |  | | --- | | <html> <body> <script type="text/javascript"> var d = new Date() document.write(d.getUTCHours()) document.write(".") document.write(d.getUTCMinutes() + 1) document.write(".") document.write(d.getUTCSeconds()) </script> </body> </html> |   [Display weekday](http://www.w3schools.com/js/tryit.asp?filename=tryjs_date_weekday) A simple script that allows you to write the name of the current day instead of the number. Note that the array object is used to store the names, and that Sunday=0, Monday=1 etc.   |  | | --- | | <html> <body> <script type="text/javascript"> var d = new Date() var weekday=new Array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday") document.write("Today is " + weekday[d.getDay()]) </script> </body> </html> |   [Display full date](http://www.w3schools.com/js/tryit.asp?filename=tryjs_date_fulldate) How to write a complete date with the name of the day and the name of the month.   |  | | --- | | <html> <body> <script type="text/javascript"> var d = new Date() var weekday=new Array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday") var monthname=new Array("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec") document.write(weekday[d.getDay()] + " ") document.write(d.getDate() + ". ") document.write(monthname[d.getMonth()] + " ") document.write(d.getFullYear()) </script> </body> </html> |   [Display time](http://www.w3schools.com/js/tryit.asp?filename=tryjs_time) How to display the time on your pages. Note that this script is similar to the Time example above, only this script writes the time in an input field. And it continues writing the time one time per second.   |  | | --- | | <html> <body> <script type="text/javascript"> var timer = null  function stop() { clearTimeout(timer) }  function start() { var time = new Date() var hours = time.getHours() minutes=((minutes < 10) ? "0" : "") + minutes var seconds = time.getSeconds() seconds=((seconds < 10) ? "0" : "") + seconds var clock = hours + ":" + minutes + ":" + seconds document.forms[0].display.value = clock timer = setTimeout("start()",1000) } </script> </body> </html> | |

**JavaScript String Object**

**The Most Common Methods**

**NN**: Netscape, **IE**: Internet Explorer, **ECMA**: Web Standard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Methods** | **Explanation** | **NN** | **IE** | **ECMA** |
| length | Returns the number of characters in a string | 2.0 | 3.0 | 1.0 |
| indexOf() | Returns the index of the first time the specified character occurs, or -1 if it never occurs, so with that index you can determine if the string contains the specified character. | 2.0 | 3.0 |  |
| lastIndexOf() | Same as indexOf, only it starts from the right and moves left. | 2.0 | 4.0 |  |
| match() | Behaves similar to indexOf and lastIndexOf, but the match method returns the specified characters, or "null", instead of a numeric value. | 4.0 | 4.0 |  |
| substr() | Returns the characters you specified: (14,7) returns 7 characters, from the 14th character. | 4.0 | 4.0 |  |
| substring() | Returns the characters you specified: (7,14) returns all characters between the 7th and the 14th. | 2.0 | 3.0 | 1.0 |
| toLowerCase() | Converts a string to lower case | 2.0 | 3.0 | 1.0 |
| toUpperCase() | Converts a string to upper case | 2.0 | 3.0 | 1.0 |

**Examples**

[The length method](http://www.w3schools.com/js/tryit.asp?filename=tryjs_length)  
The length method returns the number of characters in a string.

|  |
| --- |
| <html> <body> <script type="text/javascript"> var str="Web Enabling Tools is Cool!" document.write("<p>" + str + "</p>") document.write("str.length") </script> </body> </html> |

[The indexOf() method](http://www.w3schools.com/js/tryit.asp?filename=tryjs_indexof)  
Test if a string contains a specified character. Returns an integer if it does and -1 if it does not. Use this method in a form validation.

|  |
| --- |
| <html> <body> <script type="text/javascript"> var str="Web Enabling Tools is Cool!" var pos=str.IndexOf("Enabling") if (pos>=0) { document.write("School found at position: ") document.write(pos + "<br>") } else { document.write("Enabling not found!") } <p>This example tests if a string contains a specified word. If the word is found it returns the position of the first character of the word in the original string. Note: The first position in the string is 0! </script> </body> </html> |

[The match() method](http://www.w3schools.com/js/tryit.asp?filename=tryjs_match)  
Works similar to the indexOf method, only this method returns the characters you specified, "null" if the string does not contain the specified characters.

|  |
| --- |
| <html> <body> <script type="text/javascript"> var str="Web Enabling Tools is cool!" document.write(str.match("cool")) </script> <p>This example tests if a string contains a specified word. If the word is found it returns the word! </body> </html> |

[The substr() method](http://www.w3schools.com/js/tryit.asp?filename=tryjs_substr)  
Returns a specified part of a string. If you specify (3,6) the returned result string will be from the third character and 6 long. (Note that since the first character is 0, the second is 1 etc, the result will be from the second character and 6 long).

|  |
| --- |
| <html> <body> <script type="text/javascript"> var str="W3Schools is great!" document.write(str.substr(2,6)) document.write("<br><br>") document.write(str.substring(2,6)) </script> <p>The substr() method returns a specified part of a string. If you specify (2,6) the returned string will be from the second character (start at 0) and 6 long. <p>The substring() method also returns a specified part of a string. If you specify (2,6) it returns all characters from the second character (start at 0) and up to, but not including, the sixth character. </body> </html> |

[The substring() method](http://www.w3schools.com/js/tryit.asp?filename=tryjs_substring)  
Returns a specified part of a string. (3,6) returns the characters from the third to the 6th.

|  |
| --- |
| <html> <body> <script type="text/javascript"> var str="W3Schools is great!" document.write(str.substring(3,6)) </script> <p>This example returns all the characters from the third character up to but not including the 6th character. (Note that since the first character is 0, the second is 1 etc, the result will be from the fourth character including the 6th character). </body> </html> |

[The toLowerCase() and toUpperCase() methods](http://www.w3schools.com/js/tryit.asp?filename=tryjs_lowerupper)  
Converts a string to lower case and upper case respectively.

|  |
| --- |
| <html> <body> <script type="text/javascript"> var str=("Hello JavaScripters!") document.write(str.toLowerCase()) document.write("<br>") document.write(str.toUpperCase()) </script> </body> </html> |