JavaScript: Objects.

In JavaScript predefined Objects

- *Strings objects*
- *Dates objects*
- *Maths objects*
- *Window Object*
- *Booleans objects*
- *Numbers objects*
- *Document objects*

String Objects

Normally, strings are created as primitives: **var firstName = "John"**

But strings can also be created as objects using the **new** *keyword:* **var firstName = new String("John")**

Number Objects

Normally, numbers are created as primitives: **var x = 123**

But numbers can also be created as objects using the **new** *keyword:* **var x = new Number(123)**

Boolean Objects

Normally, booleans are created as primitives: **var x = false**

But booleans can also be created as objects using the **new** *keyword:* **var x = new Boolean(false)**

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
constructor	*Returns a reference to the String function that created the object.*
length	*Returns the length of the string.*
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 demonstrate the usage of String properties.

String Methods

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

Method	Description
charAt()	*Returns the character at the specified index.*
concat()	*Combines the text of two strings and returns a new string.*
indexOf()	*Returns the index within the calling String object of the first occurrence of the specified value, or -1 if not found.*
lastIndexOf()	*Returns the index within the calling String object of the last occurrence of the specified value, or -1 if not found.*
match()	*Used to match a regular expression against a string.*

replace()	*Used to find a match between a regular expression and a string, and to replace the matched substring with a new substring.*
search()	*Executes the search for a match between a regular expression and a specified string.*
split()	*Splits a String object into an array of strings by separating the string into substrings.*
substr()	*Returns the characters in a string beginning at the specified location through the specified number of characters.*
substring()	*Returns the characters in a string between two indexes into the string.*
toLowerCase()	*Returns the calling string value converted to lower case.*
toString()	*Returns a string representing the specified object.*
toUpperCase()	*Returns the calling string value converted to uppercase.*

Example on String methods:

```
<head>
<script>
var st=new String("test java prg java script java lang");
var st1=new String("TEST");
var st2=new String("JAVA");
document.write(st);document.write("<br>");
document.write(st.substr(5,8)+"<br>"); //java prg
document.write(st.substring(5,8)+"<br>");// jav
document.write(st.indexOf("java")+"<br>"); // 5
document.write(st.indexOf("java",8)+"<br>");   //14
document.write(st.toUpperCase()+"<br>"); // TEST JAVA PRG JAVA SCRIPT JAVA LANG
document.write(st.small()+"<br>");// test java prg java script java lang
document.write(st.big()+"<br>");
document.write(st.concat(st1,st2)); //test java prg java script java langTESTJAVA
document.write("<br>");
ptr=st.split(" ");
document.write("<br>");
document.write(ptr.length); // 7
document.write("<br>");
for(i=0;i<ptr.length;i++)
document.write("<br>"+ptr[i]);
ptr.sort();
for(i=0;i<ptr.length;i++)
document.write("<br>"+ptr[i]);
add=ptr.join(" ");
document.write("<br>"+add);

</script>
</head>
```

The Date Object

Syntax

You can use any of the following syntaxes to create a Date object using Date() constructor.

```
new Date( )
new Date(milliseconds)
new Date(date string)
new Date(year,month,date[,hour,minute,second,millisecond ])
```

- **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. For compatibility (in order to avoid the Y2K problem), 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.*

Date Methods

Here is a list of the methods used with **Date** *and their description.*

Method	Description
Date()	Returns today's date and time
getDate()	Returns the day of the month for the specified date according to local time.
getDay()	Returns the day of the week for the specified date according to local time.
getFullYear()	Returns the year of the specified date according to local time.
getHours()	Returns the hour in the specified date according to local time.
getMilliseconds()	Returns the milliseconds in the specified date according to local time.
getMinutes()	Returns the minutes in the specified date according to local time.
getMonth()	Returns the month in the specified date according to local time.
getSeconds()	Returns the seconds in the specified date according to local time.
getTime()	Returns the numeric value of the specified date as the number of milliseconds since January 1, 1970, 00:00:00 UTC.
getYear()	**Deprecated** - Returns the year in the specified date according to local time. Use getFullYear instead.

setDate()	Sets the day of the month for a specified date according to local time.
setFullYear()	Sets the full year for a specified date according to local time.
setHours()	Sets the hours for a specified date according to local time.
setMilliseconds()	Sets the milliseconds for a specified date according to local time.
setMinutes()	Sets the minutes for a specified date according to local time.
setMonth()	Sets the month for a specified date according to local time.
setSeconds()	Sets the seconds for a specified date according to local time.
setTime()	Sets the Date object to the time represented by a number of milliseconds since January 1, 1970, 00:00:00 UTC.
setYear()	**Deprecated -** Sets the year for a specified date according to local time. Use setFullYear instead.
toString()	Returns a string representing the specified Date object.
valueOf()	Returns the primitive value of a Date object.

Program to display digital clock:

Program to display digital clock:

```
<html>
<head>
<script type="text/javascript">
function setTime()
{
var today=new Date()
var h=today.getHours()
var m=today.getMinutes()
var s=today.getSeconds()
txt.innerText=h+":"+m+":"+s

}
function startTime()
{
setInterval('setTime()',500)
}
</script></head>

<body onload="startTime()" >
<h1 align="center"> <font color="red">
<div id="txt"></div>
</font></h1>
</body>
</html>
```

The Math Object

Syntax

The syntax to call the properties and methods of Math are as follows

```
var pi_val = Math.PI;
var sine_val = Math.sin(30);
```

Math Methods

Here is a list of the methods associated with Math object and their description

Method	Description
abs()	*Returns the absolute value of a number.*
acos()	*Returns the arccosine (in radians) of a number.*
asin()	*Returns the arcsine (in radians) of a number.*
atan()	*Returns the arctangent (in radians) of a number.*
atan2()	*Returns the arctangent of the quotient of its arguments.*
ceil()	*Returns the smallest integer greater than or equal to a number.*
cos()	*Returns the cosine of a number.*
exp()	*Returns E^N, where N is the argument, and E is Euler's constant, the base of the natural logarithm.*
floor()	*Returns the largest integer less than or equal to a number.*
log()	*Returns the natural logarithm (base E) of a number.*
max()	*Returns the largest of zero or more numbers.*
min()	*Returns the smallest of zero or more numbers.*
pow()	*Returns base to the exponent power, that is, base exponent.*
random()	*Returns a pseudo-random number between 0 and 1.*

round()	*Returns the value of a number rounded to the nearest integer.*
sin()	*Returns the sine of a number.*
sqrt()	*Returns the square root of a number.*
tan()	*Returns the tangent of a number.*
toSource()	*Returns the string "Math".*

Methods of document object

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

The important methods of document object are as follows:

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.*

Window Object:

The window object represents an open window in a browser.

If a document contain frames (<iframe> tags), the browser creates one window object for the HTML document, and one additional window object for each frame.

Methods: window.prompt() – create prompt dialog box

Window.open(“url”) – open given URL in browser

Wimdow.alert() – create alert dialog box

Wondow.confirm() –create confirm dialog box,