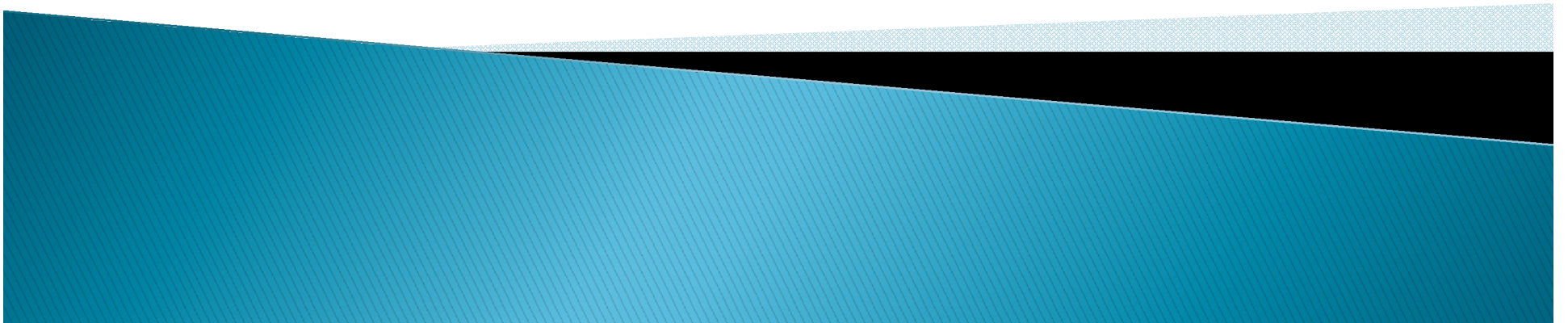


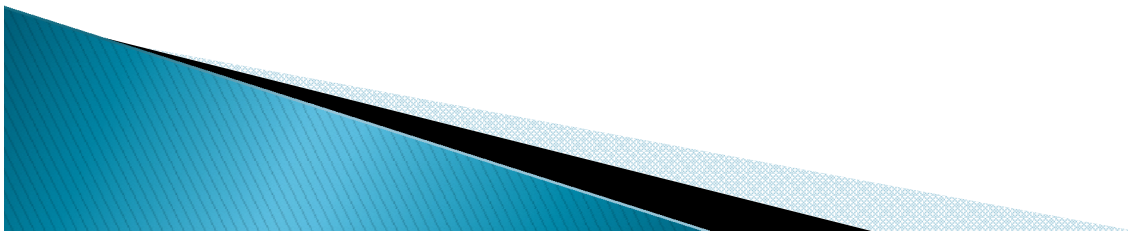
# JavaScript

## Objects and Events



# Content

- ▶ Introduction
- ▶ Types
- ▶ Examples

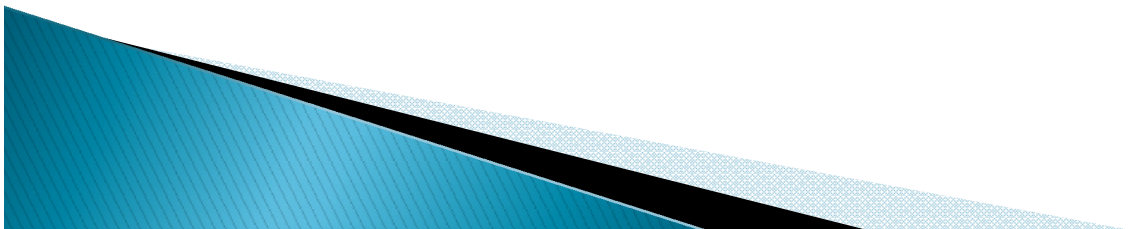


# INTRODUCTION

- ▶ By using JavaScript, we have the ability to create dynamic web pages. Events are actions that can be detected by JavaScript.

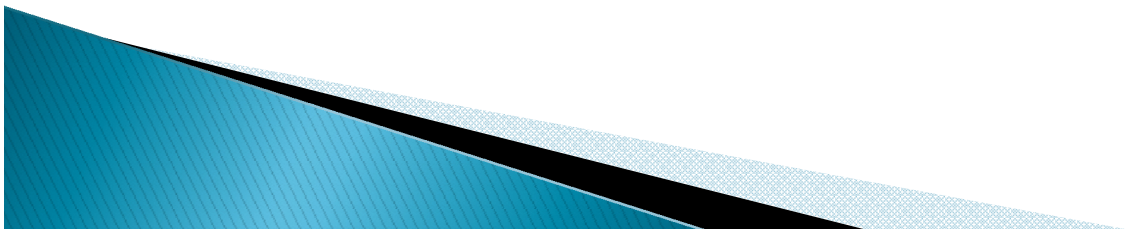
Examples of events:

- ▶ A mouse click
- ▶ A web page or an image loading
- ▶ Mousing over a hot spot on the web page
- ▶ Selecting an input box in an HTML form
- ▶ Submitting an HTML form



# *onload and onUnload*

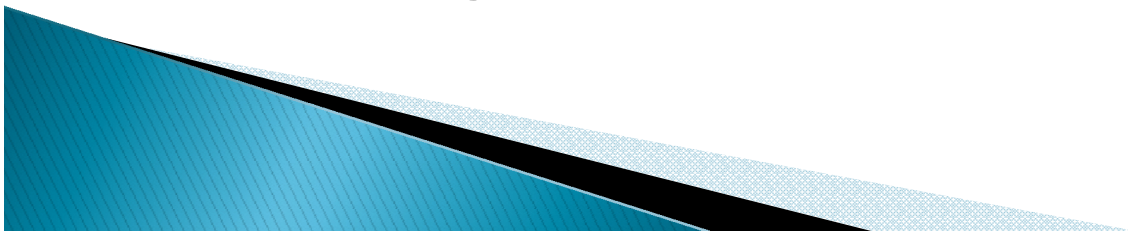
- ◉ The onload and onUnload events are triggered when the user enters or leaves the page.
- ◉ The onload event is often used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.
- ◉ Both the onload and onUnload events are also often used to deal with cookies that should be set when a user enters or leaves a page.



# *onFocus, onBlur and onChange*

- The onFocus, onBlur and onChange events are often used in combination with validation of form fields.
- Below is an example of how to use the onChange event. The checkEmail() function will be called whenever the user changes the content of the field:

```
<input type="text" size="30" id="email"  
onchange="checkEmail()" >;
```



# *onSubmit*

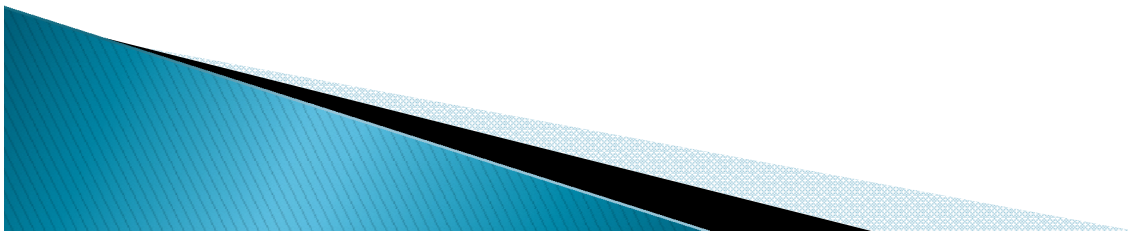
- ▶ The onSubmit event is used to validate ALL form fields before submitting it.
- ▶ Below is an example of how to use the onSubmit event. The checkForm() function will be called when the user clicks the submit button in the form. If the field values are not accepted, the submit should be cancelled. The function checkForm() returns either true or false. If it returns true the form will be submitted, otherwise the submit will be cancelled:

```
<form method="post"  
action="xxx.htm"onsubmit="return  
checkForm()" >
```



# *onMouseOver and onMouseOut*

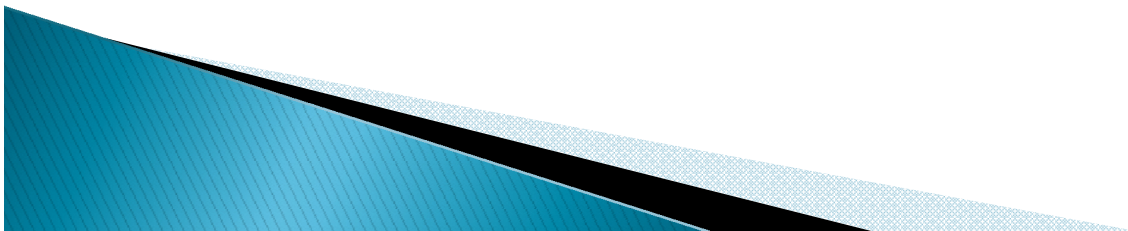
- ▶ onMouseOver and onMouseOut are often used to create "animated" buttons.
- ▶ Below is an example of an onMouseOver event. An alert box appears when an onMouseOver event is detected:



# Example

```
<html>
<head>
<script type="text/javascript">
function message()
{alert("Welcome guest!")}
</script>
</head>

<body>
<input type="button" value="View message" onclick="message()"
  />
</body>
</html>
```

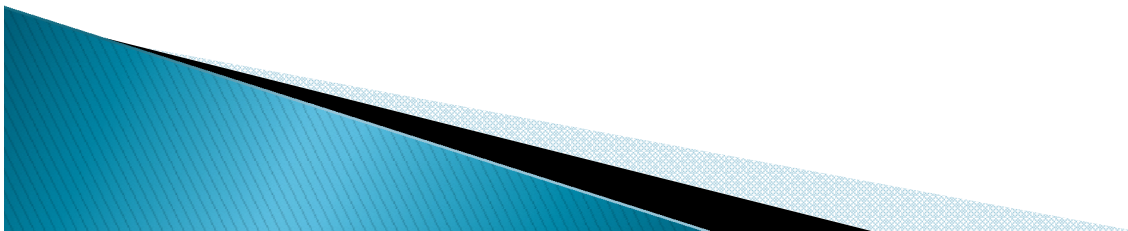




onchange	script	Script runs when the element changes
onsubmit	script	Script runs when the form is submitted
onreset	script	Script runs when the form is reset
onselect	script	Script runs when the element is selected
onblur	script	Script runs when the element loses focus
onfocus	script	Script runs when the element gets focus
onkeydown	script	Script runs when key is pressed
onkeypress	script	Script runs when key is pressed and released
onkeyup	script	Script runs when key is released
onclick	script	Script runs when a mouse click
ondblclick	script	Script runs when a mouse double-click
onmousedown	script	Script runs when mouse button is pressed
onmousemove	script	Script runs when mouse pointer moves
onmouseout	script	Script runs when mouse pointer moves out of an element
onmouseover	script	Script runs when mouse pointer moves over an element
onmouseup	script	Script runs when mouse button is released

# Object Model in JavaScript

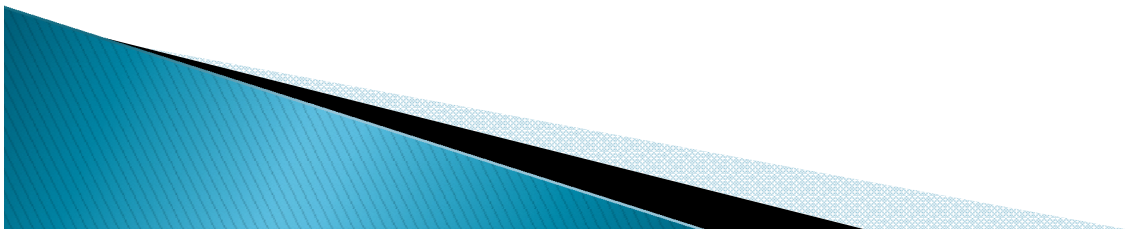
- ▶ JavaScript is an Object Oriented Programming (OOP) Language.
- ▶ Object is just a special kind of data. An object has properties and method.



# Properties

- ▶ Properties are the values associated with an object. Here in example we are using the length property of the string object to return the number of characters in a string.
- ▶ 

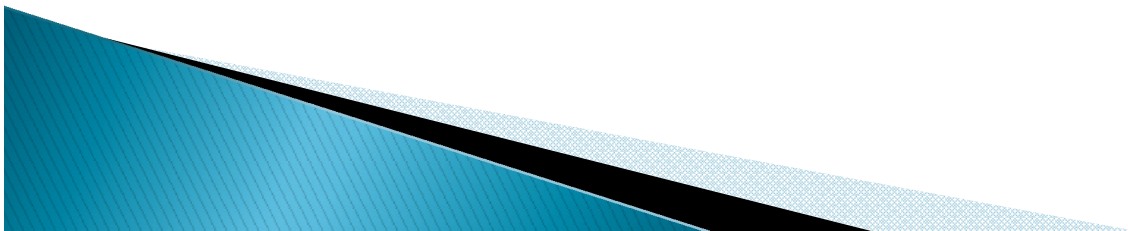
```
<script type = "text/javascript" >  
    Var text = "Hello World!"  
    Document.write (txt.length)  
</script >
```



# Methods

- ▶ Method are the actions that can be performed on objects. Here we are using the toUpperCase() method of the String object to display a text in Upper case.
- ▶ 

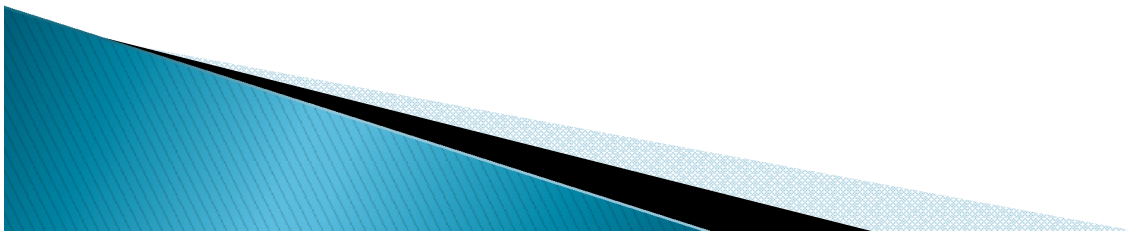
```
<script type = "text/javascript" >  
    Var str = "Hello World!"  
    Document.write (str.toUpperCase())  
</script >
```



# String object

- ▶ The string object is used to manipulate a stored piece of text.

```
txt="hello world"  
document .write(txt.length)
```

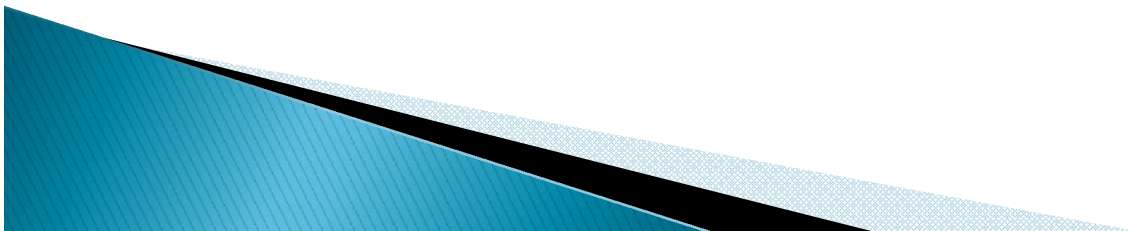


# Defining Dates

- ▶ The Date object is used to work with dates and time. we define a date object with the new keyword.

```
Var myDate=new Date()
```

- ▶ Note: The Data object will automatically hold the current date and time as its initial value.



# Date functions

Method	Description
Date()	Returns a Date object
get Date()	Returns the date of a Date object(from 1-31)
getDay()	Returns the day of a Date object(from 0 to 6, where 0 is Sunday.....)
getMonth()	Returns the month of a Date object( from 0 to 11)
getFullYear()	Returns the year of a Date object(four digits)
getYear()	Returns the year of a Date object (from 0-99)
getHours()	Returns the hour of a Date object (from 0-23)
getMinutes()	Returns the minute of a Date object( from 0-59)
getSeconds()	Returns the second of a Date object(from 0-59)

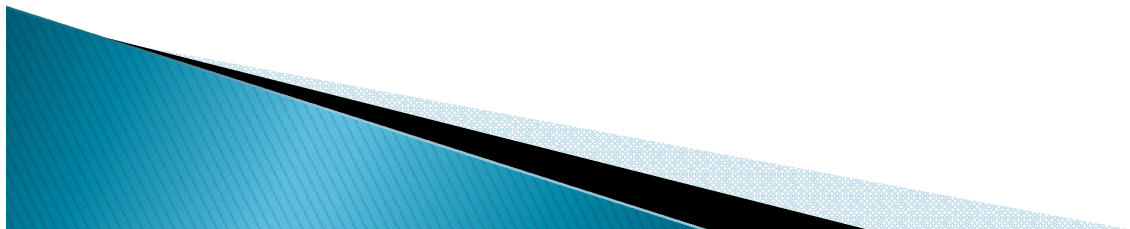
# Math functions

Method	Description
abs (x)	Returns the absolute value of x
cos (x)	Returns the cosine of x
exp (x)	Returns the value of E raised to the power of X.
log(x)	Return the natural log of X.
max (x,y)	Return the number with the highest value of x and y
min (x,y)	Return the number with the lowest value of x , y
pow(x,y)	Returns the value of the number x raised to the power
random()	Returns a random between 0 and 1 number
Sqrt(X)	Returns the Square root

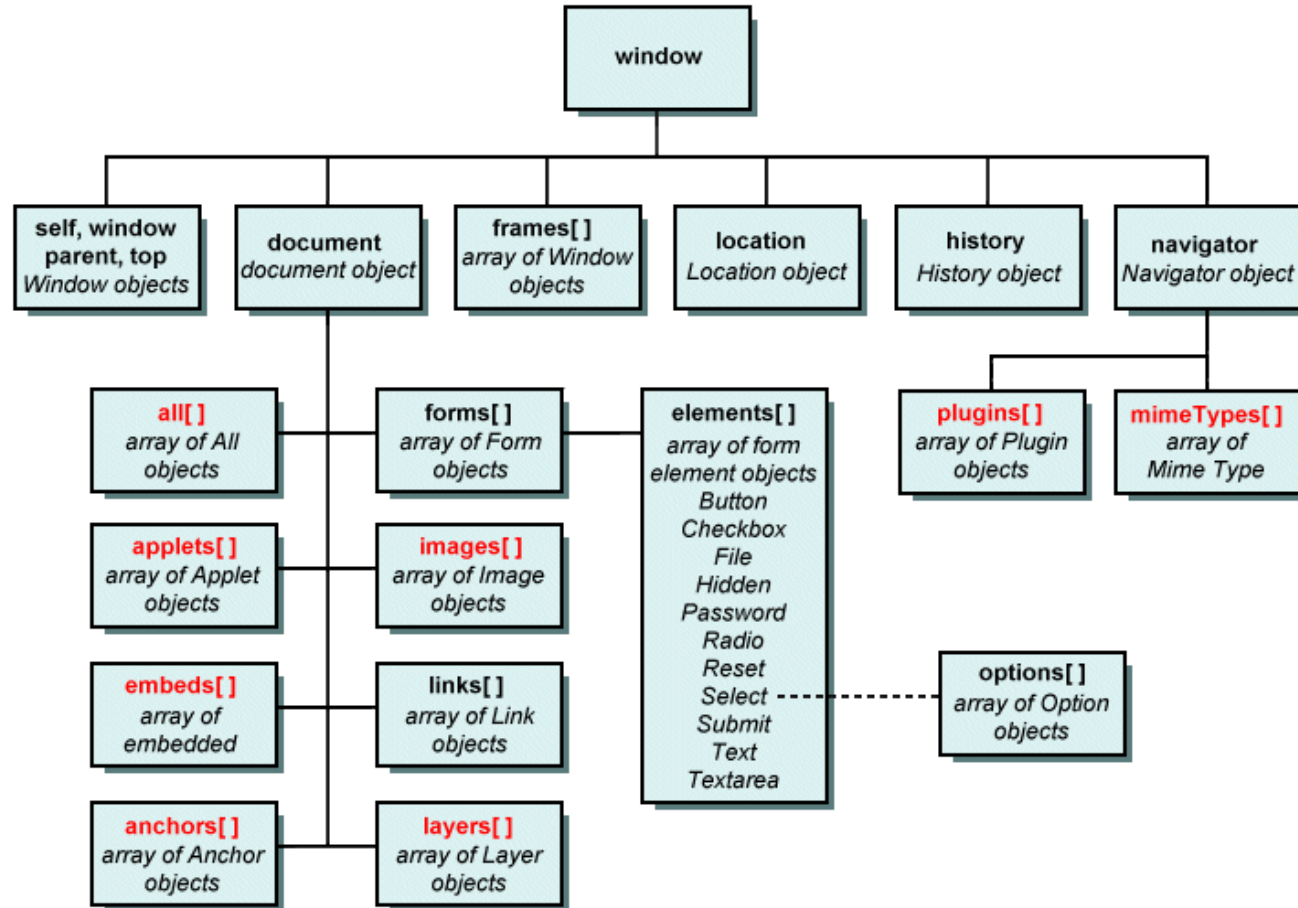


# More Java Script Object

Object	Description
Windows	The top level object in java script hierarchy . The windows object represents a browser window. A window object is created automatically with every instance of a <body> or <frameset> tag
Navigator	Contains information about the clients browser
Screen	Contains information about the client's display screen
History	Contains the visited URLs in the browser window
Location	Contains information about the current URL.



# JS Object Model



# JS functions for handling strings

- ▶ *indexOf* *0*: finds location of a specified set of characters(i.e., of a sub-string). Starts counting at 0,return starting position if found, else return -1.
- ▶ *lastIndexOf* *0*: similar to the above, but looks for the last occurrence of sub-string.
- ▶ *charAt* *0*: Returns a single character inside a string at a specific position.
- ▶ *substring* *0*: returns a sub string inside a string at a specific position.
- ▶ *split* *0*: Divides a string into a sub strings, based on delimiter.

