
JavaScript

Day 3 : Tuesday
Created Date : 21/08/2012
Created By : Alok Kumar Vishwakarma
Mentor : Satheesh Balasubramanian

- What is JavaScript?
- JavaScript Basic Elements
- JavaScript Objects

What is JavaScript?

JavaScript is an implementation of **ECMAScript** language standard.

ECMA Standard: The ECMA standard is also called as ECMAScript-262 was approved as an international ISO (ISO/IEC 16262) standard in 1998. ECMA-262 is the official JavaScript standard.

Purpose and usage of JavaScript

- JavaScript was designed to add interactivity to the HTML pages.
- To do client-side validation of form input.
- To detect the visitor's browser.
- To create cookies.
- To manipulate HTML elements.

JavaScript Basic Elements

JavaScript Statement: A JavaScript statement is a command to browser.

Purpose of JavaScript Statement: To tell the browser what to do.

Example:

```
document.getElementById("demo").innerHTML="Hello JavaScript" ;
```

JavaScript Comments:

there is two types of comments in javascript.

1. Single Line comment
2. MultiLine comment

SingleLine comment starts with //.

example:

//write to a heading

```
document.getElementById("demo").innerHTML="Hello JavaScript" ;
```

MultiLine comment starts with /* and end with */

```
/* This
```

```
is
```

```
MultiLine comment */
```

```
document.getElementById("demo").innerHTML="Hello JavaScript" ; // write to a heading
```

JavaScript Variables:

variables are "containers" for storing information.

Rules:

* variable names are case sensitive (x and X are two different variables).

** variable names must begin with a letter, the \$ character or the underscore character.

Variable Declaration in JavaScript:

```
var userId;  
userId="008";
```

Points to Remember when dealing with JavaScript variables-

1. When you assign a text values to a variable put quotes around the value.
2. when you assign a numeric value to a variable, do not put quotes around a numeric data, it will be treated as text.
3. if you re-declare a javascript variable, It will not lose its value.

Types of JavaScript Variables:

1. Local Variables
2. Global Variables.

Local Variables: A variable declared within a javascript function becomes LOCAL and can only be accessed within that function.

Note:

* we can have local variables with the same name in different functions, because local variables are only recognized by the function in which they are declared.

** local variables are deleted as soon as the function is completed.

Global Variables: A variable declared outside a function becomes GLOBAL and all scripts and functions on the web page can access it.

Note:

* Global variables are deleted when you close the page.

JavaScript Operators:

= assignment operator used to assign values.

+ used to add values.

Arithmetic Operators: +, -, *, /, %, ++, --

Assignment Operators: =, +=, -=, *=, /=, %=

Note:

* To add two or more string variables or text values the + operator can be used.

** Adding two numbers will return the sum but adding a number and a string will return a string.

Comparison Operators: ==, === (is exactly equal to (value and type)), !=, !== (is not equal to (neither value or type)), >, <, >=, <=

Logical Operators: &&, ||, !

Conditional Operator: ?

Syntax:

variablename = (condition) ? value1: value2 ;

example:

voteable=(age<18) ? "Too young" : "Old enough" ;

Conditional Statements:

1. if statement
2. if-else statement
3. if...elseif...else statement
4. switch statement

JavaScript Popup Boxes:

In javascript there are 3 types of popup boxes.

1. Alert Box
2. Confirm Box
3. Prompt Box

Alert Box: It is used if n we want to make sure information comes through to the user. when an alert box pops up the user will have to click "OK" to proceed.

Syntax: alert("message to be displayed");

Confirm Box: It is used if we want the user to verify or accept something.

Syntax: confirm("message to be displayed");

Prompt Box: It is often used if we want the user to input a value before entering a page.

Syntax: prompt("some message","default-value");

LineBreaks: To display line breaks inside a popup box, use a backslash by the character n.

Example: alert("Hello \n How are you ");

JavaScript Functions: A function is a block of code that executes only when you tell it to execute.

Syntax:

```
function function_name()
{
    // javascript code here
}
```

Types of JavaScript functions:

1. function with argument

```
function myFunction( var1, var2)
{
// java script code here
}
```

2. function with a return value

```
function myFunction( )
{
var x=5;
return x;
}
```

JavaScript Loop:

There is two different kind of loops.

1. for loop- loops through a block of code a specified number of times.

Syntax:

```
for (var=start_value; var< end_value; var= var+ increment )
{
    // code to be executed
}
```

2. while- loops through a block of code while a specified condition is true.

Syntax:

```
while( var < end_value)
{
    // code to be executed
}
```

do-while loop syntax:

```
do {
    // code to be executed
}
while(var < end_value);
```

Break Statement: The break statement will break the loop and continue executing the code that follows after the loop.

Continue Statement: The continue statement will **break the current iteration** and continue the loop with the next value.

JavaScript for...in statement : The for...in statement loops through the properties of an object.

Syntax: for (variable in object)

```
{
    // code to be executed
}
```

example:

```
var person={fname:"john",lname: "Doe",age:25};
for (x in person)
{
    txt=txt + person [x];
}
```

JavaScript Events: Events are actions that can be detected by javascript.
for example: clicking a button.

onload and onunload : These events are triggered when the user enters or leaves the page.

example: <body onload="checkcookies()" />

onfocus, onblur and onchange : The onfocus, onblur and onchange events are often used in combination with validation of form fields.

example:

```
<input type="text" onchange= "checkEmail()" />
```

onsubmit : The onsubmit event can be used to validate form fields before submitting it.

example: <form method= "post" action="mySubmit.jsp" onsubmit="checkForm()" />

onmouseover, onmouseout: The onmouseover and onmouseout events can be used to trigger a function when the user mouses over or out of an HTML element. there are other events such as onmousedown, onmouseup and onclick.

try...catch Statement : It allows us to test a block of code for errors.

```
try {  
    // some code  
}  
catch(err)  
{  
    // handle errors here  
}
```

throw statement: It allows us to create an exception.

Syntax: throw exception ;

The exception can be a string, integer, boolean or an object.

JavaScript special characters: we can add special characters using backslash sign.

```
\ ' single quote  
\" double quote  
\\ backslash  
\n new line  
\r carriage return  
\t tab  
\b backspace  
\f form feed
```

JavaScript Objects

- JavaScript is an object based programming language.
- An object based programming language allows us to define our own objects and make our own variable types.

Properties: Properties are the values associated with an object.

Methods : Methods are the actions that can be performed on objects.

List of JavaScript Objects:

1. JavaScript String Object
2. JavaScript Date Object
3. JavaScript Array Object
4. JavaScript Boolean Object
5. JavaScript Math Object
6. JavaScript RegExp Object

1. JavaScript String Object :

The String object is used to manipulate a stored piece of text.

for example: `var txt="Hello World ";`
`document.write(txt.length);`

the o/p is: 12

Syntax: `var txt=new String ("string") ;`
or `var txt="string" ;`

String object properties:

constructor - returns the function that created the string object's prototype.

length - returns the length of a string.

prototype - allows you to add properties and methods to an object.

String Object Methods :

- `charAt()`
- `charCodeAt()`
- `concat()`

- fromCharCode()
- indexOf()
- lastIndexOf()
- match()
- replace()
- search()
- slice()
- split()
- substr()
- substring()
- toLowerCase()
- toUpperCase()
- valueOf()

String HTML wrapper Methods :

- anchor()
- big()
- blink()
- bold()
- fixed()
- fontcolor()
- fontsize()
- italics()
- link()
- small()
- strike()
- sub()
- sup()

2. JavaScript Date Object :

The date object is used to work with dates and times.

Create a Date Object: The date object is used to work with dates and times.

Date objects are created with the date constructor. there are four ways of initiating date.

- new Date() // current date and time
- new Date(milliseconds) // milliseconds since 1970/01/01
- new Date(dateString)
- new Date(year, month, day, hours, minutes, seconds, milliseconds)

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

Set Dates: We can easily manipulate the date by using the methods available for the Date object.

example: var myDate=new Date();
 myDate.setFullYear(2010,0,14);

Compare two Dates : The Date object is also used to compare two dates.

Date Object Properties:

constructor - Returns the function that created the date object's prototype.

prototype - Allows us to add properties and methods to an object.

Date Object Methods :

- getDate()
- getDay()
- getFullYear()
- getHours()
- getMilliseconds()
- getMinutes()
- getMonth()
- getSeconds()
- getTime()
- getTimezoneOffset()
- getUTCDate()
- getUTCDay()
- getUTCFullYear()
- getUTCHours()
- getUTCMilliseconds()
- getUTCMinutes()
- getUTCMonth()
- getUTCSeconds()
- parse()
- setDate()
- setFullYear()
- setHours()
- setMilliseconds()
- setMinutes()

- setMonth()
- setSeconds()
- setTime()
- setUTCDate()
- setUTCFullYear()
- setUTCHours()
- setUTCMilliseconds()
- setUTCMinutes()
- setUTCMonth()
- setUTCSeconds()
- toDateString()
- toISOString()
- toJSON()
- toLocaleDateString()
- toLocaleTimeString()
- toLocaleString()
- toString()
- toTimeString()
- toUTCString()
- UTC()
- valueOf()