

**[ADD THIS PAGE TO THE NAVIGATION STRUCTURE TO DISPLAY A PAGE BANNER HERE]**

Last Edit: 14-Sep-09

More On This  
[Add this page to the Navigation view to display hyperlinks here]  
Apart From This  
[Add this page to the Navigation view to display hyperlinks here]

Abhinav Junior College

Sub: - Information Technology

Std: -XII

2004-2005

## Introduction to JavaScript

### Netscape and JavaScript

JavaScript is a cross platform scripting language (web site development environment) created by Nets with the Netscape suite of Client and Server products.

The Netscape client 'browser' product is called Netscape Communicator. The default scripting language understands is JavaScript.

One Netscape server product, from its suite of server products, is Netscape Commerce Server. The Netscape Commerce Server understands is JavaScript.

### JavaScript

JavaScript is an object-oriented language that allows creation of interactive Web Pages. JavaScript allow into an HTML form to be processed as required. This empowers a web site to return site information acc

JavaScript offers several advantages to a Web developer. A short development cycle. Ease of learning JavaScript can be easily and quickly used to extend the functionality of HTML pages, already on a web s

### The advantages of JavaScript

- **An interpreted language** JavaScript is an interpreted language, which requires no compilation. The process. The syntax is completely interpreted by the browser just as it interprets HTML tags.
- **Embedded within HTML** JavaScript does not require any special or separate editor for programs to can be written in any text editor like Notepad, along with appropriate HTML tags, and saved a embedded JavaScript commands can then be read and interpreted by any browser that is JavaScript
- **Minimal Syntax – Easy to Learn** By learning just a few commands and simple rules of syntax, comp using JavaScript.
- **Quick Development** Because JavaScript does not require time-consuming compilations; scripts ca of time. This is enhanced by the fact that many GUI interface features, such as alerts, prompts elements, are handled by client side JavaScript, the browser and HTML code.
- **Designed for Simple, Small Programs** It is well suited to implement simple, small programs The web page.
- **Performance** JavaScript can be written such that the HTML files are fairly compact and quite requirements on the web server and download time for the client. Additionally, because, JavaScript in the same file as the HTML code for a web page, they require fewer separate network accesses.
- **Procedural Capabilities** Every programming language needs to support facilities such as Co: Branching. JavaScript provides syntax, which can be used to add such procedural capabilities to we
- **Designed for Programming User Events** JavaScript supports Object / Event based programin form 'Button' is pressed. This event can have suitable JavaScript code attached, which will execute occurs. JavaScript can be used to implement context sensitive help. Whenever an HTML form's button or a link on the page a helpful and informative message can be displayed in the status l window.
- **Easy Debugging and Testing** Being an interpreted language, JavaScript scripts are tested line by line they are encountered, i.e. an appropriate error message along with the line number is listed for ev thus easy to locate errors, make changes, and test it again without the overhead and delay of compil

- **Platform independence / Architecture Neutral** JavaScript is a programming language that is hardware on which it works. It is a language that is understood by any JavaScript enabled browser work on any machine that has an appropriate JavaScript enabled browser installed.

### Writing JavaScript into HTML

JavaScript syntax is embedded into an HTML file. A browser reads HTML files and interprets HTML tags. To be included as an integral part of an HTML document when required, the browser needs to be informed that the code is JavaScript. The browser will then use its built-in JavaScript engine to interpret this code.

The browser is given this information using the HTML tags `<SCRIPT>...<SCRIPT>`. The `<SCRIPT>` tag marks the start of scripting code. The paired `</SCRIPT>` marks the end of the snippet of scripting code.

Like most other HTML tags, the `<SCRIPT>` tag takes in an optional attribute, as listed below:

Attributes	Description
Language	Indicates the scripting language used for writing the snippet of scripting code. If left undefined Netscape Communicator will assume JavaScript. If left undefined Internet Explorer will assume VB Script.

Syntax :

```
<SCRIPT LANGUAGE= "JavaScript">
```

```
    JavaScript code snippet written here
```

```
</SCRIPT>
```

**Statement** is a line of code. This has to be ended with a semicolon(;).

**Block** is a set of statements put into a pair of brackets.({,})

**Comments:** Single line comments starts by //.

Multi line comments start with /\* and end with \*/

### JavaScript Data Types and Literal

JavaScript supports four primitive types of values and supports complex types such as arrays and objects. A variable can be assigned a single literal value such as number, string or Boolean value. Literals are fixed values, used throughout a program.

The primitive data types that JavaScript supports are:

#### **Number:**

JavaScript supports all numbers the same. They are just numbers whether they are integers or floating point numbers (with decimal places)

eg.10,21.9,0.764

#### **Boolean:**

Consists of the logical value true and false. JavaScript supports a pure Boolean type that consists of two values: true and false. Boolean operators can be used in Boolean expressions. JavaScript automatically converts the Boolean values to numbers when they are used in numerical expressions. Values 1 and 0 are not considered Boolean values in JavaScript.

**String:**

Consists of string values that are enclosed in single or double quotes. JavaScript provides built-in sequence of one or more characters that are enclosed by double (") or single (') quotes. If a string begins with a double quote. If a string begins with a single quote it must end with a single quote.

Example: "Rahul", '24, Sanjay Nagar, Bangalore '

If a string has to include quote character in the string the quote character must be preceded by the backslash.

**Operators & Expressions**Arithmetic operators:

Common operators are addition(+), subtraction (-), multiplication (\*), division (/), and Modulus (%). These are binary operators which need two operands.

- sign is used as a unary negation operator. This helps to negate the sign of the value.

Post increment (x++), pre increment (++x)

post decrement (x--), pre decrement (--x). These are some unary operators used to increase or decrease the value.

```
var x=10;
```

```
y=++x; (x value incremented and assigned to y)====>y is 11,x is 11---->pre increment
```

```
var x=10;
```

```
y=x++; (x value assigned to y and then incremented)====>y is 10,x is 11---->post increment
```

```
var x=10;
```

```
y--x; (x value decremented and assigned to y)====>y is 9,x is 9---->pre decrement
```

```
var x=10;
```

```
y=x--; (x value assigned to y and then decremented)====>y is 10,x is 11---->post decrement
```

Logical operators

and -->&& exp1 && exp2 true if exp1 and exp2 are true

or -->|| exp1 || exp2 true if either exp1 is true or exp2 is true

not -->! !exp1 true if exp1 is not true

Comparison operators

Equals x==y true if x is equal to y


not equal x!=y true if x is not equal to y

less than x<y true if x is less than y

less than or equal x<=y true if x is less than or equal to y

greater than x>y true if x is greater than y

greater than or equal x>=y true if x is greater than or equal to y

 **Creating Variables**

In order to make working with data types convenient, variables are created. In JavaScript variables can hold any data type of data.

In order to use a variable, it is good programming style to declare it. Declaring a variable tells JavaScript to exist so that the JavaScript interpreter can understand the difference to that variable name throughout the rest of the program.

Although it is possible to declare the variables by simply using them, declaring variables helps to organize and helps keep track of the scope of the variables. Variables can be declared using the var keyword.

Syntax:

```
Var <variable name>=value;
```

Examples

```
Var first_name;
```

```
Var last_name="Shah";
```

```
Var phone=6128879;
```

The equal sign (=) used in assigning a value to a variable is known as an assignment operator. Like programming languages, variable names are case sensitive.

Variable name should not be a reserved word, should start with a character and can be followed by alphanumeric characters.

**Functions.** A function is a set of instructions which can be identified by a unique name and supposed to receive values and they can return a value although it is not compulsory.

Functions in JavaScript are of two types .

1) Inbuilt functions & 2) User Defined Functions

Inbuilt functions are part of JavaScript. They are ready to use functions for a programmer who writes code.

eg. eval(), isNaN(), prompt(), confirm(), alert()

User defined functions .These are the temporary functions written by the programmer for a particular purpose. A function is defined by combining a function statement and a block of code to associate with that function. A function is defined by followed by the name of the function and a set of parenthesis .

### **sample program**

```
<html>
<head>
<title>JavaScript function</title>
</head>
<body>
<h1>this HTML page demonstrates how to write a function</h1>
<script language=JavaScript>
function printtext(thestring) // this is definition of a user defined function
{
thestr=eval(thestring) //eval() is a built in functions
document.write ("used built-in eval function to evaluate");
document.write ("the converted value is");
document.write (thestr);
}
var convertme="100+200";
printtext(convertme);
</script>
</body>
</html>
```

Data can be returned from a function by using the keyword return.

data can be returned from a function by using the keyword return.

```
function square(innm)
{
return (innm*innm)
}

<script language = "JavaScript">

var squared=square(4);

document.write (squared); // The output will be 16.

</script>
```

**Flow control**

Different programming constructs are sequence, selection and Iteration.

Sequence---> Statements executed one after another.

Selection--->Conditional execution of different statement or block of statements .

*if, if---else, switch*

Iteration---> Repetition of the statements for a definite number of times or on a conditional basis.

*for,while,do--while*

Flow interruption statements are 1)*break* & 2)*continue*

*break*- allows the control to come out of a loop.

*continue*- allows to continue the loop skipping the current execution of the statements which are below it

**Objects:** Java script is an object based scripting language. One of the features of OOP (Object Oriented Programming) is Encapsulation functions and variables are wrapped into a single unit which can be used in the form of objects. Properties and the functions are called methods.

Methods in an object can not be called as such. Function call should be prefixed with the name of the object.

eg. mycar.left(); mycar is the object,left() is a method.This causes the car to turn left.

eg. mycar.speed=20; speed is a property of mycar object. 20 is the value.

Some commonly used objects are date,string,document,form,button,text,textarea etc

date: methods---getDate(),getDay(),getHours().....

string: property---length  
property---toUpperCase(),toLowerCase(),substring()...

document: property---title,bgcolor,fgcolor.....  
method---write(),writeln(),close(),open().....

form: properties---name,length,method,action.....  
method---reset(),submit()

**text:** properties--- value,defaultvalue,form,name  
methods--- blur(),focus(),select()

**textarea:** properties--- value,defaultvalue,form,name  
methods--- blur(),focus(),select()

**Arrays:** array is a collection of values kept one after another.

arrays are objects in javascript.

```
var myarray=new Array();
```

or

```
var myarray=new Array(5);
```

Index starts with zero.The last index is length-1.Javascript does not insists all data in the array

need to be of the same type.

```
var arr1=new Array("hi",5,false);
```

The number of subscripts in the array can be found out by arrayname.length property.

### **Events and Event handlers.**

**Event** is the action recognized by the object to which it responds. Events are the mechanism by which I This enables you to develop web pages that are more interactive, more responsive, and easier to use.

For example, when a user clicks a hyperlink or a button, or enters data in a form, an event is generated i action has occurred and that further processing is required. The browser waits for events to occur, and w whatever processing is assigned to those events. The processing that is performed in response to the occu: **event handling**. The code that performs this processing is called an **event handler**.

JavaScript defines events for most of the major objects found in Web pages including links, images, form JavaScript mainly there are two types of event exists and they are *keyboard events* and *mouse events*.

### **Communicating with the User:**

Web pages are filled with information for users, but most of it is static. You can use scripts to display dyn

#### Displaying Information from Client Scripts

Because client scripts run on the browser, they give you flexibility in how you want to display informati display message boxes, the way stand-alone applications on a computer often do.

Some of the most useful methods provided by the Window object are those that support dialog with the follows:

The **alert()** method displays a dialog box containing a message and an OK button. You use the alert() m critical information that must be acknowledged, by means of an OK button.

Syntax : alert (text)

The **confirm()** method is similar to the alert() method except that it produces a dialog box with a messag button. The confirm() method returns *true* if the user clicks OK and *false* if the user clicks Cancel. You use the user and ask him to confirm whether he wants to perform a particular action.

Syntax : confirm (text)

The *prompt()* method displays a message to the user and prompts the user to type information into a text field. You use the *prompt()* method to obtain text input from the user. The value entered by the user is the return value of the *prompt* method.

Syntax : `prompt (text, default Input)`

Example : `-alert()`

Example :- `confirm()`

Example :- `prompt()`

#### **Example of alert()**

```
<html>
<head> <title> Window Object </title> </head>
<body>
<script language="JavaScript">
window.alert ("You are doing JavaScript-Alert program!!");
</script>
</body>
</html>
```

#### **Example of confirm()**

```
<html>
<head> <title> Window Object </title> </head>
<body>
<script language="JavaScript">
window.confirm ("Do you want save the changes ?");
</script>
</body>
</html>
```

#### **Example of prompt()**

```
<html>
<head> <title> Window Object </title> </head>
```

```
<body>  
  
<script language="JavaScript">  
  
var name;  
  
name = window.prompt("Enter your name:"," ");  
  
alert("Welcome "+name)  
  
</script>  
  
</body>  
  
</html>
```

This site is authored and maintained by [M/s Cyber Planet](#) for [Abhinav Vidyalay & Jr. College](#). The said institution provides educational services in the area of primary, secondary & higher secondary education and thus the content is the intellectual property of the staff and students of this institution. No part of it may be reproduced without the consent of the institution.