

JAVASCRIPT WEB PROGRAMMERS GUIDE LESSON 1

File:	JSGuideL1.doc
Date Started:	Dec 14, 1999
Last Update:	May 1, 2003
ISBN:	0-9730824-0-2
Version:	2.0

LESSON 1 JAVASCRIPT PROGRAMMING

INTRODUCTION

JavaScript is a programming language similar to **Java** that is used in designing WEB pages. You need to know HTML to use Java Script. The prerequisite for this course is knowledge of HTML. If you do not know HTML then you should take our HTML programming course first. JavaScript is used to extend the capabilities of HTML not to replace HTML. The word "**script**" means the HTML program will call segments of a programming language to do a specialized task. These specialized tasks are called scripts. Since this scripting language used is similar to Java, so these scripts are then called **Java Scripts**.



Java Scripts are different from Java and are not Applets. Applets are programs written in Java and are already compiled to run on a WEB browser. Applets are usually called from HTML web pages.

Introduction to Objects

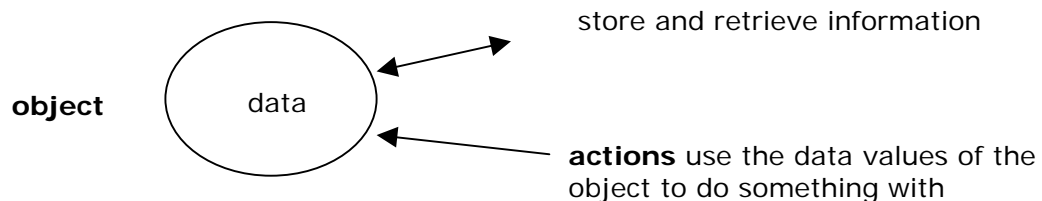
JavaScript is like Java and is also an **Object Oriented** language. When we say **objects** we mean we are going to have many things grouped together. Each thing has a value. Values may be numeric like **1234** or text like the color **blue**. A better name for thing is properties or attributes. We say a group of properties having something in common is known as an object. Each property has a value. A good example of an object is a car. A car has properties color, shape, number of wheels, number of doors etc.



The values of the properties may be numeric or text. For example the color may be blue, the shape may be sporty or smooth, the car has 4 wheels and 2 doors. What other properties of cars can you think of ? Make, Model and Year. Very good! You may have notice that cars do things, they go fast they stop etc. What a car does is known as action. Actions affect the property values where the properties describe or identify the cars. Here is some example of car actions. If you paint a car the color may no longer be blue it would be now red. Actions change the values of the properties, if a car goes fast, then its speed property will increase. If a car stops its speed property will be 0. You may have many cars all having different property values but the actions on each object will be the same. Actions can be applied to any object but only one object at a time. You need to indicate which object you are going to apply the action to.

Representing Objects in Programming

We need a programming language to represent **objects**. It would be very difficult to store a car inside a computer. I'm sure some of our students have already tried to drive their car into their computer. In a programming language the properties of an objects are represented as variables. The values of the properties are stored in the variables. The variables identify the place in the computer memory where the property values are stored. All computers have memory where they can store values. The values are also called data values or just data. Computer memory is also used to execute code. Code contains instructions to let computer do something like add two numbers, print something on the screen or get a value from the keyboard. Code is also used to perform actions on the objects. Code is used to change and use the data value of the properties of the object. You may need to do a mathematical calculation or just change or access the value of an object's property. In a programming language each object will have its own **data**, that describes the object. Each object will also have **code** associated with it so that the object can also do something. The data variables let the object store values and the code let's the object do some **action**. The **JavaScript** language has pre-defined all the objects for you. They have defined all properties of the object and actions to act upon the objects. All you got to do is use them! The data values of an object are known as **properties** and the actions are known as **methods**.



The JavaScript language has its own built in objects modeled after the HTML tags. The JavaScript objects also have their own data properties and methods. JavaScript is also an **event driven** language. This means the program will automatically respond to some event. If someone presses a button with a mouse click the program will automatically respond to that event. Your web browser must be JavaScript enabled before you can use it to run Java Scripts.

JAVASCRIPT LANGUAGE

NAMES AND VARIABLES

Variables in a programming language are used to store data values. A variable gets a name so that you can identify it. Names must start with a letter. Properties of an object are also stored in variables.

count

You may also include numbers and underscores in your variable names after the first letter.

count1
count_down

Underscores _ should be avoided it is more modern to use a capital letter to separate compound names like:

countDown

Declaring variables

In Java Script variables are declared when you want to use them. To declare a variable the **var** keyword is used followed by the variable name. All programming statements in JavaScript end with a **semicolon** ';'. We use the following color codes to identify Java scripts. The **purple text** is the definition where the **blue text** is the example and **green text** is the comment.

```
var variable_name;
```

```
var x; // declare a variable called x
```

You can also initialize a variable when it is declared:

```
var variable_name = value;
```

```
var x = 0; // declare a variable called x initialized to 0
```

The value is evaluated from the expression. The expression may be a number or another variable. Variables in JavaScript may be **local** or **global**. Global variables may be used for the whole program where local means available for use for parts in the program only. Global variables are declared at the start of the JavaScript code or declared without the **var** keyword.

```
variable_name = value;
```

```
x=0; // declare a variable called x initialized to 0
```

Local variables are declared using the **var** keyword. You must be careful to use a **var** keyword for local variables. If you do not then they become global variables and your Java script will not operate correctly.

value types

We have different type of variables because we will be working with different types of data. We will have numbers like 1234 and we will also have messages known as Strings like "hello". Messages are enclosed by double quotes. If we put double quotes around 1234 then we would have "1234" which now would be a message rather than a number. We also have a value type to represent **true** or **false** known as **boolean**. When we do not want a variable to represent a value type then we use the keyword "**null**", which means no value. The following chart lists all the value types we have discussed. Literals are used to represent the values. Literals are hard coded values and may be numbers like 1234 or letters like "hello". Literals are also known as **constants**. The following table demonstrates declaring and initializing variables.

value type	Literal	example declaring
Number	1234	var x = 1234;
Floating point	12.34 or 1.234E01	var f = 12.34;
String	"hello"	var s = "hello";
Boolean	true or false	var b = true;
null	no value	var s = null;

When you declare and initialize variable Java script automatically figures out what kind of data you want to represent.

JAVASCRIPT PRE-DEFINED OBJECTS

JavaScripts depends on predefined objects and methods of those objects. An object has **properties** to describe the object and **methods** to enable the object to do something. Data properties are represented by data values stored in the variables. Methods are represented by programming statements. Programming statements are instructions telling the computer what to do. Programming statements are executed sequentially one by one. Programming statements are grouped together in a method to do a particular task for an object. To enable a group of programming statements to execute and identified as a unit they must be grouped together in a method. The object oriented programming approach uses Objects having data and code properties.

web browser objects

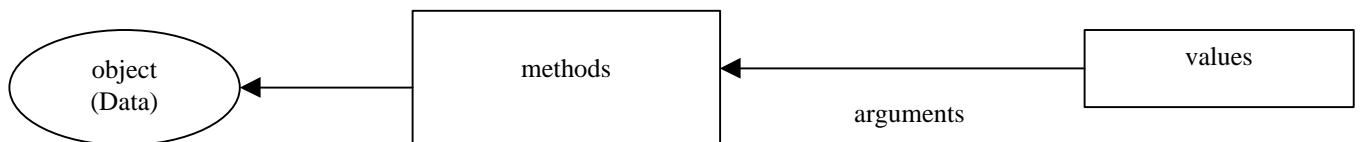
What kind of objects do we need for a web browser? One kind of object we need is one that takes care of all the things that happen on a web browser screen. The JavaScript object that does this is known as the "**document object**". What can this object do ? One thing we would like to do is to write a message on the web browser screen. Methods use the data variables of an object to do something. A method of the document object that can write a message to the screen is the "**write**" method. The write method contains programming statements that are executed sequentially one by one to write the specified message to the screen. When you use a method is it also referred to as calling a method. To use a method to act on an object you specify the object name followed by a "." the dot operator. You then specify name of method you want to use and the values required by the method.

object name . method_name (argument);

The dot "." operator says use this method to act on this object. The methods have round bracket that encloses the values. Each value is known as an **argument**. The argument values are sent to the method. To write a message on the web browser screen you would use the following JavaScript programming statement:

document.write ("I like JAVA script");

The write method uses the document object to write a message on the web browser screen. The string "**I like JAVA script**" is sent to the write method to be displayed on the web browser screen. The dot operator "." is used to indicate which object the method will use. Only certain methods can be used with certain objects.



The arguments are used pass values to the methods. The methods will use these argument values to do calculations or display values. You may have more than one argument each separated by commas. When you have more than one argument then the arguments are known as an argument list.

object_name . method_name (argumen1, argument2, , argumentn);

RUNNING JAVA SCRIPTS

To run a JavaScript you include your JavaScript in your HTML file. To include a JavaScript in your HTML file, the JavaScript is included between the `<SCRIPT>` tags. When the web browser sees the script tags it will run the JavaScript between them.

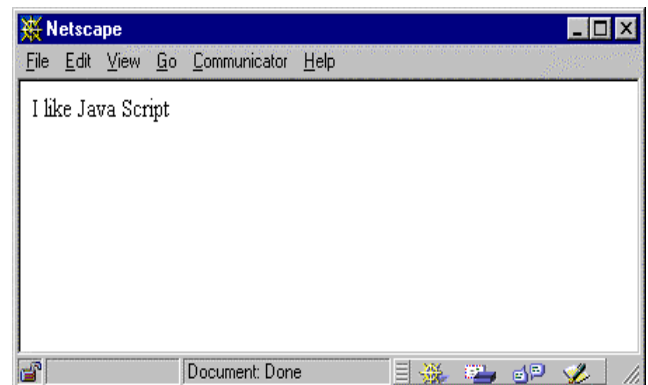
```
<script language="JavaScript">
document.write("I like JavaScript");
</script>
```

To be able to run the JavaScript it must be put into a HTML file. An HTML file starts with the `<html>` tag. The HTML file contains a header section that includes information about this HTML file. A header section starts with a `<head>` tag and ends with a `</head>` tag. The HTML file also has a body section that is used to display all the data content of the HTML file. The body section starts with a `<body>` tag and ends with a `</body>` tag. The HTML file ends with the `</html>` tag. The Web browser runs the JavaScript programming statements located inside the `<head>` tags. Comment tags are used to prevent web browsers that are not JavaScript enabled, not to display the JavaScript code on your web browser screen. If your web browser is not JavaScript enabled then it would treat the JavaScript as a message text and print it out to the screen. A comment tag starts with a `<!--` tag and the JavaScript code and ends with a `//-->` tag. Don't forget the `//` or else you will get `a -- >` on your web screen. Here is our complete JavaScript script embedded into an HTML file.

```
<!--L1p1.htm -->
<html>
<head>

  <script language="JavaScript">
    <!--
    document.write("I like JAVA script");
    //-->
  </script>

</head>
<body>
</body>
</html>
```



LESSON1 EXERCISE 1

Type the above code into an HTML file called L1ex1.htm and type in your own message. To run on your web browser just open the file and make sure your web browser is JavaScript enabled. If it is not then you will not get your message displayed on your web browser

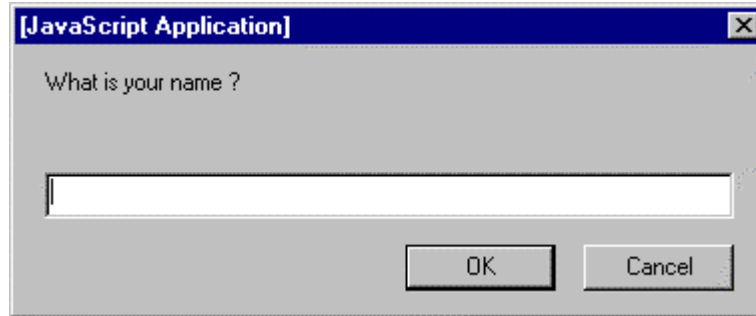
USING VARIABLES IN JAVA SCRIPT

Variables in JavaScript can be used right when you need them. The variable can represent a string or a number depending on how it is used. We can demonstrate the use of a variable by using a JavaScript **window** object and the **prompt** method of the **window object** to get an input string from the keyboard. You will learn about all of the JavaScript objects shortly. Its good to know how to use something before you know what it is.

We use the **prompt** method on a window object to get a person's name from the keyboard.

```
var name = window.prompt ("What is your name ?", "");
```

The window prompt method allows you to display a message and an optional initial value to appear in the place where you enter your name. We have set this initial value to "" which means display no value.



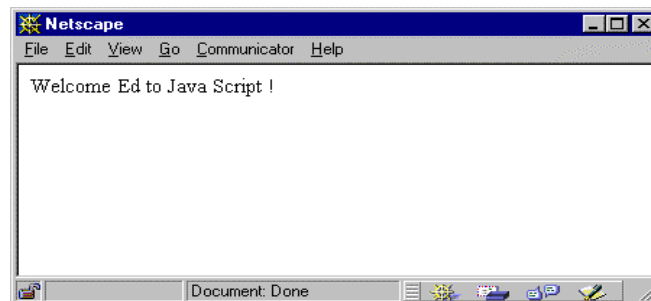
Once we get the person's name we can write it to the screen with a friendly message using the **write** method of the document object.

```
document.write ("HELLO" + name);
```

The "+" joining operator is used to join messages or messages and variables together.

Here is the complete HTML file:

```
<!--L1p2.htm -->
<html>
<head>
<script language="JavaScript">
  <!--
    var name = window.prompt("What is your name ?", "");
    document.write(" Welcome " + name + " to JavaScript !");
  //-->
</script>
</head>
<body>
</body>
</html>
```



LESSON 1 EXERCISE 2

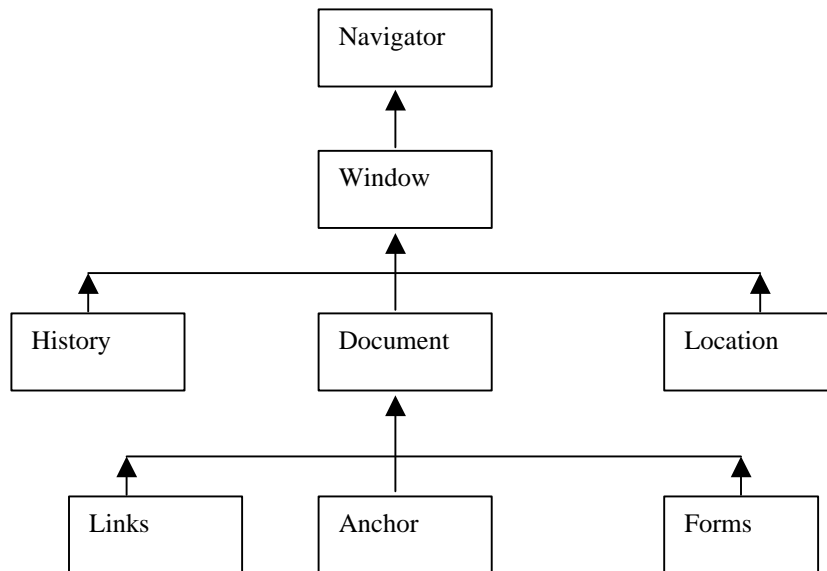
Type the above Java Script code into an HTML file and change the greeting to your own name. Run the program on your web browser. You will now be amazed how powerful JavaScript is. Already you can prompt the user for a name and then display the results. Call your program L1Ex2.htm

LESSON 1 EXERCISE 3

Write a JavaScript program that asks the user for a name, how many items they want to buy and their credit card number. Once they enter their credit card number, display a confirmation message and then ask them if the data was entered properly. Call your program L1ex3.htm

INTRODUCTION TO JAVASCRIPT OBJECTS

What Object Oriented Programming is trying to do is to represent everyday items as objects. A person is an object, a book is an object. Objects have properties. An object may have a shape or color. Objects also do things. For an example, a person object has height and weight, people eat, run sleep etc. Objects contain data and do some action. The following chart is the objects used in Java Script to represent your web page. The Navigator is the top object where all other objects descend from Navigator.



The **Navigator** object contains the **Window** object. The **Window** object contains the **History**, **Document** and **Location** objects. The document contains the **Links**, **Anchor** and **Foms** objects. We will now briefly introduce you to each object. We will study in depth of about each object in future Lessons. We already know how to use the Window and Document objects.

NAVIGATOR OBJECT

The Navigator object provides information about the WEB browser that you are using. A Navigator object has four data properties. All property names are case sensitive.

property	description		accessing
appName	returns the name of the users browser		navigator.appName
appVersion	returns version and operating system of browser in the form: release number(platform; country);		navigator.appVersion
	release number	version number of platform	
	platform	operating system used	
	country	I for international countries U for US	
appName	in house development code name		navigator.appCodeName
userAgent	used to identify the abilities of the browser		navigator.userAgent

WINDOW OBJECTS

Window objects let you have your own display area and **frame**. A **frame** is just a name for another screen on your web browser. You already know how to use the prompt method of the window object to put a message to ask the user to type in something.

```
var name = window.prompt("What is your name ?", "");
```

We will study window objects in more detail in future lessons.

History Objects

History objects keep track of all the URL's your web browser has visited per session by means of a history list. The history object has methods to go forward and backward through the history list. We will also study history objects in detail in future lessons.

Location object

The location object identifies the current document and consists of a complete **URL**. **URL** stands for **U**niform **R**esource **L**ocator and is the WEB address the document is presently accessing. An URL is defined as follows:

protocol//hostname/pathname

<http://www.cstutoring.com/index.htm>

You can also specify a starting spot by specifying an anchor location using a hash

protocol//hostname#hash/pathname

<http://www.cstutoring.com#top/index.htm>

Information can be sent to an another web page using a search string.

protocol//hostname:port/pathname/?search

<http://www.cstutoring.com:21/index.htm/?from=search&what=courses>

URL Properties

Here are the URL properties with description and examples.

property	description	example
protocol	defines the transport protocol to transfer the data	http: Hyper Text transfer Protocol ftp: File Transfer Protocol file: transfer file's mailto: sending e-mail
hostname	name of the host computer	www.cstutoring.com
port	the port being used (optional)	port 80 default
pathname	path and file name	current page to access
search	query string passes information to the web server as to search	www.cstutoring.com /?from=search&what=courses
hash	identifies an anchor in the document	www.cstutoring.com#top
host	combination of hostname and port	www.cstutoring.com:port21
href	entire URL	http://www.cstutoring.com

To see the individual parts of the current URL of the web page you are at, you can print individual location URL properties to the screen

document.write(location.protocol);

document.write(location.pathname);

To see the entire current URL of the web page you are at you can print to the screen

document.write(location);

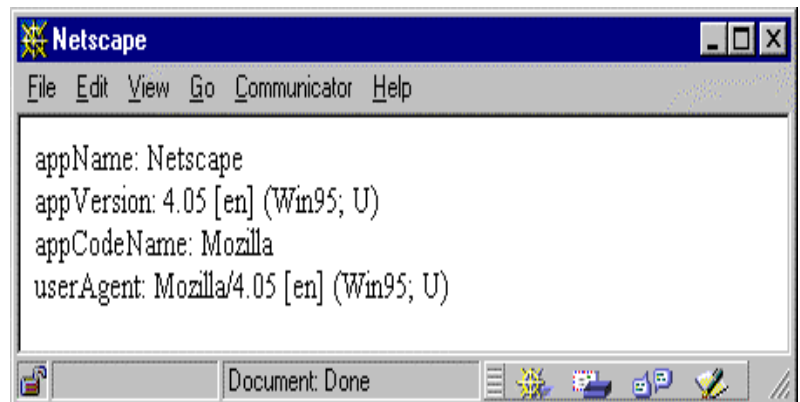
To go to another web page URL you can type

location = url;

location = "http://www.cstutoring.com";

LESSON1 EXERCISE 4

Write a JavaScript that displays all the information about your web browser. You have to use the document, navigator and location objects. Call your program L1Ex4.htm your output should be something like this:



DOCUMENT OBJECTS

The document object has all the HTML elements attached to it. Anchors, forms, history and link objects. We will study document objects and the objects attached to it in more detail in future lessons. You have already used the write method of the document object when you have written a message to the web browser screen.

```
document.write("I like JAVA script");
```

LINK and ANCHOR OBJECTS

Link and anchor objects are used to direct the user to another web page in the current web site or to another web site. Link and anchor objects are created in HTML. An Anchor represents where the web page will be directed to. An anchor is a hypertext target. The following is an anchor.

```
<a name = "anchorName" > anchor text </a>
```

```
<a name = "top" > we are at the top </a>
```

Links are used to jump to targets. Links contain a hypertext reference (href) that allow you to go to other WEB pages or parts of WEB pages in your current WEB page. Links are included inside **<a>** tags.

There are four types of links.

- (1) **links to anchors** that use the # sign and the anchor name.

```
<a href="#anchorName" > link text </a>
```

```
<a href="#top" > go to the top </a>
```

- (2) **relative links** that are links inside your WEB site

```
<a href=webpage > link text </a>
```

```
<a href="more.htm">click here to know more</a>
```

- (3) **absolute links** refer to other WEB sites.

```
<a href="url" > link text </a>
```

```
<a href="http://www.cstutoring.htm">click here to know more</a>
```

- (4) **links to anchors** in other web pages that use the # sign and the anchor name.

```
<a href="url#anchorName" > anchor text </a>
```

```
<a href="www.cstutoring.com/index.htm#top" > go to the top </a>
```

Each WEB site will have a collection of WEB pages all interconnected by links.

ANCHORS AND LINKS USING JAVA SCRIPT

You can also make anchors and links using JavaScript. You use the **String** object and the **anchor()** method of the string object to make an **anchor** in Java Script. A string object is automatically made for you when you use a string value . Strings values are text enclosed in double quotes like **"I am a string"**. The string object holds the string text.

```
string_name.anchor(anchorName);

var top = "you are at the top";

document.write(top.anchor("top"));
```

You use the String object and the **link()** method of the string object to make a **link** in Java Script.

```
string_name.link(href);

var jump = "click here to go to the top";

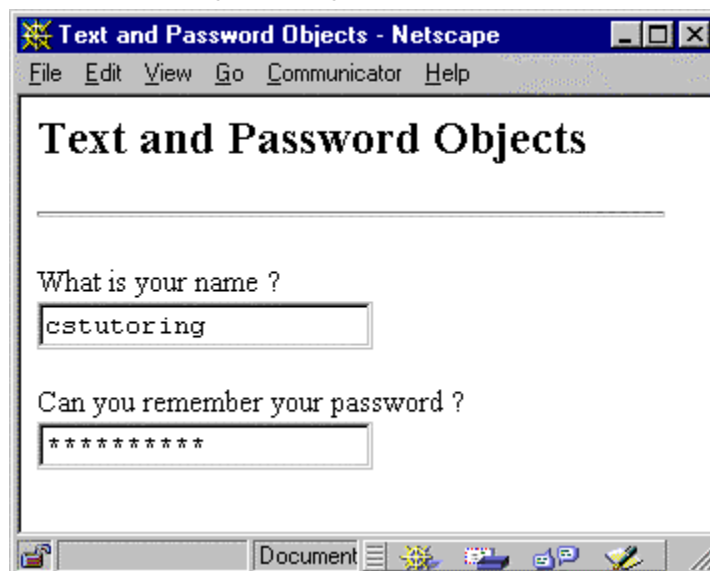
document.write(jump.link("#"+"top")+"<br>");
```

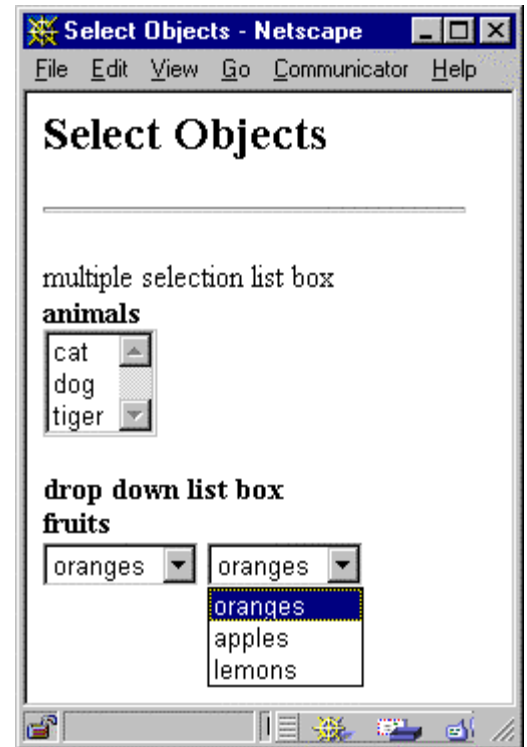
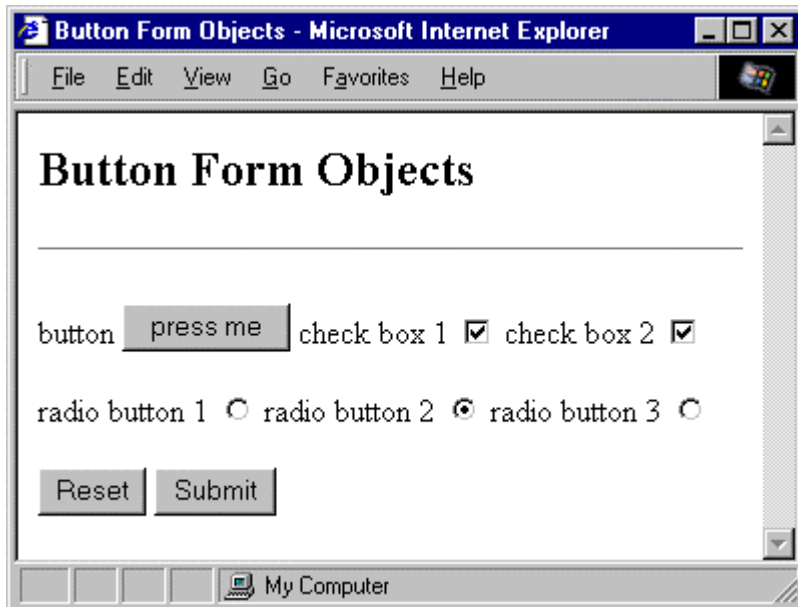
LESSON1 EXERCISE 5

Write HTML program that has a JavaScript that has an absolute or relative link to your favorite web page. Also in your HTML program put some anchors in your web page and links to those anchors. You can put the javascript inside the **<body>** tags. You may need many **
** between anchors to get the jumping effect. Call your HTML file L1Ex5.htm

FORMS OBJECTS

Form objects are created from HTML **<form>** tags. Form objects are buttons, text boxes, radio and check boxes, select list boxes and text area boxes. Form objects let the user send information back to the web page they are displaying. We will study form objects in detail in future lessons. It is good to get introduced to a topic first before you study it in depth.





IMPORTANT

You should use all the material in all the lessons to do the questions and exercises. If you do not know how to do something or have to use additional books or references to do the questions or exercises, please let us know immediately. We want to have all the required information in our lessons. By letting us know we can add the required information to the lessons. The lessons are updated on daily bases. We call our lessons the "living lessons". Please let us keep our lessons alive.

E-Mail all typos, unclear test, and additional information required to:

courses@cstutoring.com

E-Mail all attached files of your completed exercises to:

students@cstutoring.com

Order your next Lesson from:

<http://www.cscourses.com/js.htm>

This lesson is copyright (C) 1998-2003 by The Computer Science Tutoring Center "cstutoring"
This document is not to be copied or reproduced in any form. For use of student only.