JavaScript

Introduction

Table of Contents

Revised 05/02

1 Introduction

JavaScript Origins

JavaScript was released by Netscape and Sun Microsystems in 1995. However, JavaScript is not the same thing as Java.

What is JavaScript

- It is a programming language.
- It is an interpreted language.
- It is object-based programming.
- It is widely used and supported
- It is accessible to the beginner.

Uses of JavaScript

- Use it to add multimedia elements

  With JavaScript you can show, hide, change, resize images, and create image rollovers. You can create scrolling text across the status bar.

- Create pages dynamically

  Based on the user's choices, the date, or other external data, JavaScript can produce pages that are customized to the user.

- Interact with the user

  It can do some processing of forms and can validate user input when the user submits the form.

Writing JavaScript

JavaScript code is typically embedded in the HTML, to be interpreted and run by the client's browser. Here are some tips to remember when writing JavaScript commands.

- JavaScript code is case sensitive
- White space between words and tabs are ignored
- Line breaks are ignored except within a statement
- JavaScript statements end with a semi- colon ;

The SCRIPT Tag

The <SCRIPT> tag alerts a browser that JavaScript code follows. It is typically embedded in the HTML.

```
<SCRIPT language = "JavaScript">
statements
</SCRIPT>
```

SCRIPT Example

- Open "script_tag.html" in a browser.
- View the Source
- Put the cursor after <! – Enter code below → and enter the following:

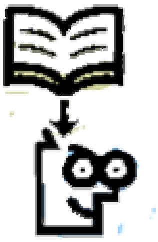
```
 <SCRIPT language = "JavaScript">
alert("Welcome to the script tag test page.")
</SCRIPT>
```

- Save the changes by choosing Save from the File menu.
- Then Refresh the browser by clicking the Refresh or Reload button.

Implementing JavaScript

There are three ways to add JavaScript commands to your Web Pages.

- Embedding code
- Inline code
- External file



External File

You can use the SRC attribute of the <SCRIPT> tag to call JavaScript code from an external text file. This is useful if you have a lot of code or you want to run it from several pages, because any number of pages can call the same external JavaScript file. The text file itself contains no HTML tags. It is call by the following tag:

```
<SCRIPT SRC="filename.js">
</SCRIPT>
```

External Example

- Open "external.html" in a browser
- View the Source
- Put the cursor after <! – Enter code here → and enter:

  ```
  <SCRIPT language = "JavaScript" SRC = "external.js">
  </SCRIPT>
  ```
- Save the changes and Refresh the browser.

2 Programming Basics

Programmers use variables to store values. A variable can hold several types of data. In JavaScript you don't have to declare a variable's data type before using it. Any variable can hold any JavaScript data type, including:

- String data
- Numbers
- Boolean values (T/F)

Variable Names

There are rules and conventions in naming variables in any programming language. It is good practice to use descriptive names for variables. The following are the JavaScript rules:

- The variable name must start with a letter or an underscore.
  firstName or _myName

- You can use numbers in a variable name, but not as the first character.
  name01 or tuition$

- You can't use space to separate characters.
  userName not user Name

- Capitalize the first letter of every word except the first
  salesTax or userFirstName

Variables

- To declare variables, use the keyword *var* and the variable name:
  var userName

- To assign values to variables, add an equal sign and the value:
  ```
  var userName = "Smith"
  var price = 100
  ```

Functions

With functions, you can give a name to a whole block of code, allowing you to reference it from anywhere in your program. JavaScript has built-in functions for several predefined operations. Here are three some functions.

- alert("message")
- confirm("message")
- prompt("message")

Function Example

- Open "functions. html" and View the Source
- Put the cursor after " // add code here " and enter:

-
  ```
  var userName
  var willDoSurvey
  userName =  prompt("Enter your name", "")
  alert("Thank you, " + userName)
  ```
- Save the changes and Refresh the page

User-Defined Functions

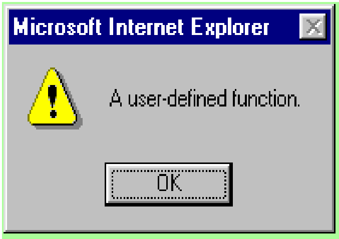
With user-defined functions, you can name a block of code and call it when you need it. You define a function in the HEAD section of a web page. It is defined with the *function* keyword, followed by the function name and any arguments.

```
function  functionName(argument)
{
statements
}
```

User-Defined Example

- Open "userdefined.html" and View the Source
- Put the cursor after "<! – enter function →" and enter:

```
<SCRIPT language = "JavaScript">
function showAlert()  {
alert("this is a user-defined function.")
}
</SCRIPT>
```



A user-defined function.

- In the BODY, put the cursor after "<! – enter the button def here ->" and enter:

```
<INPUT type-"button" value="Run the Function"
onClick="showAlert()">
```

- Save the changes and Refresh the browser.

RUN THE FUNCTION

3 Objects

JavaScript supports programming with objects. Objects are a way of organizing the variables. The different screen elements such as Web pages, forms, text boxes, images, and buttons are treated as objects.

Properties and Methods

Every object has its own properties and methods.

- Properties define the characteristics of an object.
  Examples: color, length, name, height, width

- Methods are the actions that the object can perform or that can be performed on the object.
  Examples: alert, confirm, write, open, close

Naming Objects

- Objects are organized in a hierarchy. To refer to an object use :
  objectName

- To refer to a property of an object use:
  objectName.propertyName

- To refer to a method of an object use:
  objectName.methodName()

Built-In Objects

Some of the built-in language objects of JavaScript offer more advanced operations such as:

- Math – provides for math calculations
- Date – provides date and time information
- String – provides for string manipulation

Math Object

The Math object provides methods for many mathematical calculations, including: abs(), log(), pow(), random(), round(), sqrt()

Format: `Math.method(#)`

Math Example

- Keep the "userdefined.html" file open
- Put the cursor in the BODY section and enter:

-
  ```
  <SCRIPT language = "JavaScript">
  var theNumber = 3.14
  myNum = Math.round(theNumber)
  </SCRIPT>
  ```
- Save the changes and Refresh the page

Date Object

The Date object provides the day, date, and time information.

Format: `dateObject.method()`

2002

Date Example

- Keep the "userdefined.html" file open
- Put the cursor in the BODY section and enter:

-
  ```
  <SCRIPT language = "JavaScript">
  var rightNow = new Date()
  theYear = rightNow.getFullYear()
  </SCRIPT>
  ```
- Save the changes and Refresh the page

String Object

The String object provides methods and properties for string manipulation and formatting.

Format: `stringName.method()`

String Example

- Keep the "userdefined.html" file open
- Put the cursor after the Date example and enter:

  ```
  var theString = "my name"
  var printString = theString.bold()
  var numChars = theString.length
  </SCRIPT>
  ```
- Save the changes and Refresh the page

theString = Jane Smith
printString = Jane Smith
numChars = 10

4 Document Object

The Document object represents the Web page that is loaded in the browser window, and the content displayed on that page, including text and form elements.

Document Methods

You can use the methods of the document object to work on a Web page. Here are the most common document methods:

- write() - write a string to the Web page
- open() - opens a new document
- close() - closes the document

Document Example

- Keep the "userdefined.html" file open
- Put the cursor after the String example and enter:
  ```
  document.write(myNum)
  document.write(theYear)
  document.write(printString)
  document.write(numChars)
  </SCRIPT>
  ```
- Save the changes and Refresh the page

Formatting What is Written

You will notice that the results of all four variable are printed on one line and without any spaces between the results. You can avoid this by including some formatting in your "document.write" statement.

Document Formatting Example

- Open "write.html" and View the Source
- Put the cursor after "<! – enter function →" and enter:

  ```
  <SCRIPT language="JavaScript">
  function newPage() {
  var userName = prompt("Please enter your name:", "")
  document.write("<H1>Welcome " + userName + "</H1><BR>")
  document.write("<H2>to your new home page.</H2>")
  }
  </SCRIPT>
  ```
- Put the cursor after "<! – enter the link here →" and enter:

  ```
  <A HREF="JavaScript:newPage()">Create-a-Page!</A>
  ```
- Save the changes and Refresh the page

Document Properties

Use the properties of the document object to set the colors of the page, the title and display the date the document was last modified. JavaScript has about 150 defined color words you can use or you can provide the hexidecimal RGB codes. Here are the most common document properties:

- bgColor
- fgColor
- linkColor
- vlinkColor
- title
- lastModified

Document Example

- Keep the "write.html" file open
- Put the cursor after the last "document.write" and enter:

  ```
  document.bgColor="red"
  ```
- Save the changes and Refresh the page

Window Object

The window object represents the browser window. You can use it to open a Web page in a new window and to set the attributes for the window. There are only two main window properties. They are:

- status - set the status bar message
- self - stores the name mof the current window

Window Methods

The window methods are mainly for opening and closing new windows. The following are the main window methods. They are:

- alert() - to display a message box
- confirm() - to display a confirmation box
- prompt() - to display a prompt box
- open() - to open a new window
- close() - to close a window

Window Example

- Open the "window.html" file and View the Source
- Put the cursor after "<! – Enter the function here →" and enter:
  ```
  <SCRIPT language = "JavaScript">
  function openWin() {
  window.open("windowtoo.html")
  }
  </SCRIPT>
  ```
- Put the cursor after "<!—Add link here →" and enter:
  ```
  <A HREF="JavaScript:openWin()">New Window!</A>
  ```
- Save the changes and Refresh the page

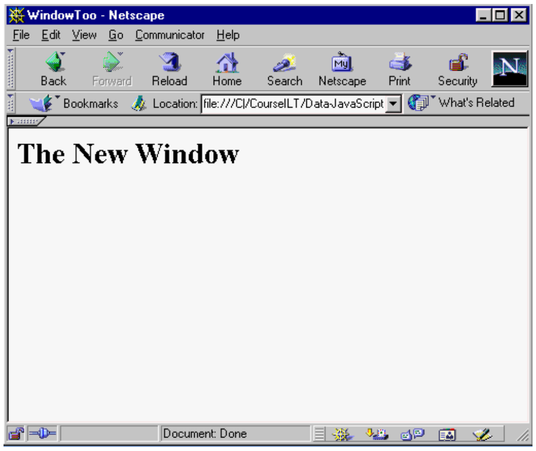
Window Attributes

If the default new window does not suit your needs, you can specify different features of the window when you open it. The complete syntax of the "window.open" is as follow:

```
window.open(URL, windowName, featureList)
```

By default, if you do not specify any features, then a window will have all of them. If you specify any one feature, then the window will have only those you set equal to 1. The following are the window attributes:

- toolbar
- menubar
- scrollbars
- resizeable
- status
- location
- directories



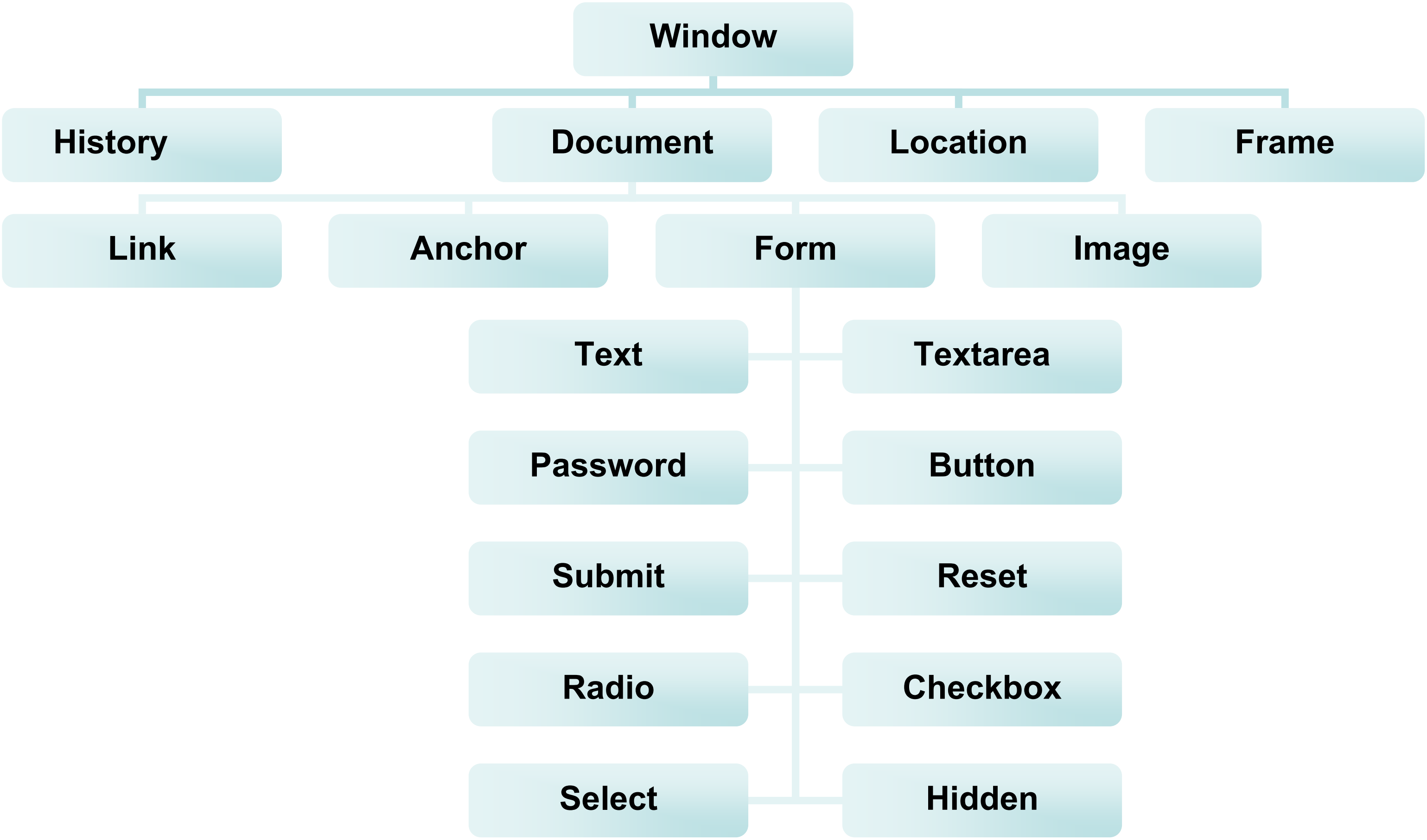
The New Window

Window Attributes Example

- With the "window.html" file open, View the Source
- Put the cursor on the line "window.open " and edit it to:
  window.open("windowtoo.html", "newWindow", "height=200,width=200,")
- Save the changes and Refresh the page

5 Objects and Events

The JavaScript Object Hierarchy



Events

The objects in a Web pages are organized in a hierarchy. All objects have properties and methods. In addition, some objects also have "events". Events are things that happen, usually user actions, that are associated with an object. The "event handler" is a command that is used to specify actions in response to an event. Below are some of the most common events:

- onLoad - occurs when a page loads in a browser
- onUnload - occurs just before the user exits a page
- onMouseOver - occurs when you point to an object
- onMouseOut - occurs when you point away from an object
- onSubmit - occurs when you submit a form
- onClick - occurs when an object is clicked

Events and Objects

Events are things that happen, actions, that are associated with an object. Below are some common events and the object they are associaated with:

Event	**Object**
onLoad	Body
onUnload	Body
onMouseOver	Link, Button
onMouseOut	Link, Button
onSubmit	Form
onClick	Button, Checkbox, Submit, Reset, Link

Example: `<FORM onSubmit="functionName()">`

Image Rollover

Replacing one image with a second image when the user moves the mouse over it is called a "rollover". The events called for are the "onMouseOver" and "onMouseOut". The object used with these events can be a link or a button.

Image Rollover Example

- With the "window.html" file open, View the Source
- Put the cursor after the line "<A HREF=" and enter:
  ```
  <P>
  <A HREF="URL"
  onMouseOver="document.hot.src='hot1.gif'"
  onMouseOut="document.hot.src='hot2.gif'">
  <IMG name="hot" src="hot2.gir"> </A>
  ```
- Save the changes and Refresh the page.