

JavaScript: A “Quick and Dirty” Overview

**Stanford University Continuing Studies CS 22
Enhanced Web Site Design**

Table of Contents

Table of Contents.....	2
Introduction to JavaScript	3
Why use JavaScript?	3
Scope	3
Next Steps	3
Overview	4
General Terms.....	5
Object Terms	6
Statement Terms	8
Resources	12

Introduction to JavaScript

Developed by Netscape, JavaScript is the easiest way to write small scripts for specific functions. The scripts are written directly into the HTML code and can be used to enhance both the look and overall functionality of your Web site.

Why use JavaScript?

JavaScript is a simple programming language that allows web developers to add interactivity and improve the look and feel of a web page. It is used to code events, objects, and actions, thus creating client and server Internet applications. As an interpreted language, there's nothing to download, and it is not computer specific. It will work as long as your system has a JavaScript-capable browser (such as *Internet Explorer*, *Firefox*, or *Netscape*) to interpret the code.

JavaScript is commonly used to create image rollovers, pop-up windows, forms, clocks, calculators, and a variety of other dynamic objects and functions. It can also be used to perform an action, such as playing an audio file or displaying a pop-up text box.

Note: Support for JavaScript varies among different browser brands and versions. For best results, use the latest version of the browser.

Scope

This is designed to be a simple overview of JavaScript and to familiarize students with the terms and concepts used in creating and editing JavaScript programs.

After this section, you will be able to:

- Understand what JavaScript programs can do
- Understand how to edit existing JavaScript programs
- Understand how to obtain JavaScript programs that others have created and use them in your own web pages

Next Steps

This is not designed to teach you everything you will need to know to create your own JavaScript programs. The next steps in training could include:

- Taking a class in JavaScripting at a local community college or computer institute
- Reading books on JavaScripting
- Viewing online training and tutorials

For some suggestions, see the Resources section at the end of this handout.

Overview

JavaScript can be placed anywhere in the HTML as long as it stays within its tags:

```
<!--  
  <script language="JavaScript">  
  
    code code code  
  
  </script>  
  
  //-->
```

However, to ensure that functions aren't accidentally called before the JavaScript is loaded, the script should go within the `<head>` and `</head>` tags.

The comment tag (`<!--` and `-->`) indicates the actual JavaScript. Everything between the `<!--` and the `-->` will be ignored by browsers that cannot process the script.

The code `//` tells the browser to ignore the following text and to exclude it from the JavaScript.

The following code is used to indicate a function:

```
function someFunction ()  
  
  {JavaScript statements }
```

This is where the actual JavaScript goes. The brackets `{ }` separate the statements in the function.

To end a JavaScript, the following code is used:

```
</SCRIPT>  
  
  //-->
```

General Terms

Action:

This property is a reflection of the action attribute in an HTML `<form>` tag, including the forms destination URL.

Cookie:

An object containing client information that can be accessed by the server

Event Handler:

Attributes of HTML tags embedded in documents. The attribute assigns a JavaScript command or function to execute when the event occurs

Function:

A user-defined or built-in set of statements that perform a task. It can also return a value when used with the return statement.

Java:

An object-oriented, platform-independent programming language developed by Sun Microsystems used to provide additional functionality to Web pages. Programming in Java requires a Java Development Kit with compiler and core classes.

Literal:

An absolute value not defined by a variable. Examples: 1, 3.1415927, "Bob", false, true.

Method:

A function assigned to an object. For example, `bigString.toUpperCase()`; returns an uppercase version of the string contained in `bigString`.

Object:

A construct with properties that are JavaScript variables or other objects. Functions associated with an object are known as the object's *methods*.

Property:

Describes an object. A property is defined by assigning it a value. There are several properties in JavaScript that contain constants (values that never change).

Protocol:

Returns a string with the initial portion of the URL, which indicates the access method (`http://`, `ftp://`, `mailto:`, etc.).

Script:

One or more JavaScript commands enclosed in a `<script>` tag.

Object Terms

Button:

A form element that must be defined within a `<form>` tag used to perform an action

Checkbox:

A form element that the users sets to "on" or "off" by clicking. It must be defined in a `<form>` tag.

Date:

Replaces a normal date type.

Document:

Created by the browser when a page containing information on the current document is loaded. These properties are defined within the `<body>` tag.

Elements:

Parts of a form (buttons, checkboxes, radio buttons, text, textarea, hidden, etc.)

Form:

A property of the document object. Each form in a document is a separate and distinct object that can be referenced using the form object. The form object is an array created as forms are defined through HTML tags. If the first form in a document is named `OrderForm`, then it could be referenced as `document.OrderForm` or `document.forms[0]`.

Frame:

A window that contains HTML sub-documents that are independently scrollable

Link:

A location document. In addition to providing information about hypertext links, the link object can be used to define new links.

Location:

Contains complete URL information for the current document, while each property of location contains a different portion of the URL

Math:

Includes properties for mathematical constants and methods for functions

Password:

Created by HTML, password text fields are masked when entered by the user

Reset:

Correlates with an HTML reset button, which resets all form objects to their default values.

Select:

A selection list or scrolling list on an HTML form.

String:

A series of characters defined by double or single quotes. For example:
`myCat="Domestic Short Haired"`

Submit:

Causes the form to be sent to the CGI program specified in the ACTION attribute.

Text:

A one-line input field.

Textarea:

Similar to a text object, with the addition of multiple lines. A textarea object can also be updated by assigning new contents to its value.

Window:

Created by the browser when a web page is loaded containing properties that apply to the whole page. It is the top-level object for each document, location, and history object. Because its existence is assumed, you do not have to reference the name of the window when referring its objects, properties, or methods.

Statement Terms

Break:

Terminates the current `for` or `while` loop and passes control to the first statement after the loop.

For:

Creates a loop with three optional expressions enclosed in parentheses and separated by semicolons, followed by a set of statements to be executed during the loop:

```
for(initialExpression; condition; updatedExpression)
{
    statements
}
```

The first expression is used to initialize the counter variable, which can be a new variable declared with the `var` (see *Var*, below). The condition expression is evaluated on each pass through the loop. If the condition is true, the loop statements are executed. The update expression is used to increment the counter variable.

```
for(variable in object)
{
    statements
}
```

For each property, the statement block is executed.

Function:

Establishes a JavaScript function with a name and parameters. To return a value, the function must include a `return` statement. A function definition cannot be nested within another function.

```
Function ([parameter][..., parameter])  
  
    {  
  
        statements  
  
    }
```

If...else:

A conditional statement that executes the first set of statements if the `condition` is true, and the statements following `else` if the statements are false.

```
if (condition)  
  
    {  
  
        statements  
  
    }  
  
else  
  
    {  
  
        statements  
  
    }
```

Return:

Specifies the value to be displayed or acted upon when the function is completed.

```
return expression;
```

Var:

Declares a variable and optionally initializes it to a value. The scope of a variable is the current function, or, when declared outside a function, the current document.

```
Var variableName=value, variableName2=value2
```

While:

Repeats a loop as long as an expression is true.

```
while (condition)
{
    statements
}
```

With:

Establishes a default object for a set of statements. Any property references without an object are assumed to use the default object.

```
with (object)
{
    statements
}
```

The `with` statement is useful when applied to the `Math` object for a set of calculations. For example:

```
with (Math)
{
    var Value1 = cos(angle);
    var Value2 = sin(angle);
}
```

would replace:

```
{
    var Value1 = Math.cos(angle);
    var Value2 = Math.sin(angle);
}
```

Resources

Web Tutorials:

- Online courses (and hundreds of books) on JavaScript are available to the Stanford community at TechPort (SUNet ID required):
<http://techport.stanford.edu>
- JavaScript resources on the Web:
<http://echoecho.com/javascript.htm>
<http://www.webmonkey.com/programming/javascript/>

Web Sites Offering Free JavaScripts:

- <http://www.javascriptsource.com/>
- <http://www.free-javascripts.com/>
- <http://www.java-scripts.net/>
- <http://www.javascriptcity.com/scripts/>

You can also download JavaScripts from the Adobe Exchange:

<http://www.adobe.com/cfusion/exchange/>

Books:

- *JavaScript: The Complete Reference* by Thomas Powell (McGraw Hill)
ISBN: 0-07-225357-6
- *JavaScript: A Beginner's Guide* by John Pollock (Osborne)
ISBN: 0-07-213140-3
- *HTML and JavaScript Basics* by Karl Barksdale and E. Turner (Thompson: Course Technology)
ISBN: 0-619-05991-5