Introduction to

www.wvu.edu/~support/training/classmat/js/

Instructor:
Kathy Fletcher
Kathy.Fletcher@mail.wvu.edu

Office of Information Technology
Customer Support

OIT Help Desk: 293-4444 x1

Table of Contents

What Is JavaScript

JavaScript is a simple, relatively easy to use programming language for Web pages. It gives you a way to add interactivity to your Web pages. With JavaScript you can transform your Web pages from static displays to pages that react to and process information from those who view your pages. JavaScript does not require a lot of programming skill on your part to get started. You can pick up the basics of JavaScript in a short time.

History of JavaScript

JavaScript was released by Netscape in 1995 under the name LiveScript. Netscape's purpose for developing LiveScript was to extend the capabilities of static HTML pages and to offload data processing from busy Web servers onto local computers running Web browsers. After JAVA was released by Sun Microsystems and gained recognition, the name LiveScript was changed to JavaScript.

Microsoft recognized the importance of JavaScript and entered the arena with two creations, JScript and VBscript. JScript is roughly compatible with JavaScript. VBscript is a subset of Visual Basic.

These competing languages have created problems for Web developers. Microsoft, Netscape, and others finally agreed to support a vendor-neutral standard. The European Computer Manufacturers Association (ECMA) produced this standard in July 1997. Today, none of the languages are 100% compliant with this standard.

Why Use JavaScript

- JavaScript offers a programming language for Web pages that most anyone can use.
- JavaScript is becoming a standard for Web page programming. This means you're more assured that your work will not soon disappear. There are thousands of Web sites that use JavaScript today.
- JavaScript makes your Web pages come alive by responding to things a user does on your pages.
- JavaScript is ideal for Form Validation. Sometimes you may want your users to enter a specific type of data (e.g. a phone number) into a form field. If it's important that this data conform to a certain format, you can use JavaScript to validate the data on the user's machine before it is forwarded to the server.
- JavaScript can open and close new browser windows and you can control the appearance of the new windows you create. You can control their size, their location, and the toolbars they have available.
- JavaScript can perform mathematical computations.
- JavaScript can make your Web pages look "fresh" and up-to-date. For example, if your Web page displays the current date, it feels a little more up-to-date.
- JavaScript can make your Web pages look ***cool***. It is easy to get carried away with this one, so the best advice is "don't overdo it."
- Learning to program in JavaScript can give you valuable experience in the important technology area of Object Oriented Programming (OOP).
- JavaScript is FUN. There are a lot of neat things you can do with JavaScript which provide immediate visual feedback.

These are just some of JavaScript's many capabilities.

JavaScript vs. Java

Don't confuse JavaScript with Java. Because of the similarity in their names, it is natural to assume that JavaScript and Java are a lot alike, but this is not the case. Other than the fact that both have the ability to deliver content over the Web, they have little else in common.

JavaScript	Java
Interpreted	Compiled
Object-Based	Object-Oriented
Cannot stand alone. Embedded in and requires HTML.	Stands alone. Java is a complete development environment.
Developed by Netscape.	Developed by Sun Microsystems.
Loosely typed	Strongly typed

NOTE: The symbol ***** is used throughout this handout. It directs you to the Web site: **www.wvu.edu/~support/training/classmat/js/**, where there are online examples prepared for this workshop. Following the * is a phrase that indicates which online example you should look at.

What Can You Do With JavaScript

You have probably run into JavaScript while using the Web. Here are some sites where it is used. You might want to visit them to see some of the things JavaScript can do.

Rollovers	pds.jpl.nasa.gov/
Pulldown Menu	www.wvnet.edu/sitemap.shtml
Online Quiz	www.wvnet.edu/~sbandyop/ECON54/mckmt1a.htm
Color Picker	www.projectcool.com/developer/reference/color-chart.html
RGB to Hex Converter	amby.com/tools/rgb2hex.html
Calculators	www.thusness.com/bmi.t.html javascript.internet.com/calculators/advanced.html
Smart PopUp Windows	www.terraincognita.com/smart/

*__What Can You Do With JavaScript.__ View these links online.

Scripting in JavaScript

JavaScript code is an extension of HTML. It appears as text right alongside HTML code.

- JavaScript appears within the <HEAD> and <BODY> sections of an HTML document.
- White spaces and line breaks mean nothing to HTML. How much space you leave between words or lines doesn't matter to HTML. It matters to JavaScript.
- Case sensitivity is unimportant to HTML. It matters to JavaScript.

Knowing all the ways in which JavaScript is embedded in HTML is important before you begin. Here are three ways you can connect JavaScript code to your HTML documents.

1. The **<SCRIPT>** and **</SCRIPT>** tags are the main way of connecting JavaScript code to HTML. You are free to embed as many scripts into a single HTML document as you like using multiple <SCRIPT> tags.

Script tags	Line	What it does
<SCRIPT LANGUAGE="JavaScript">	5	tells your browser that JavaScript code is about to appear.
</SCRIPT>	23	tells your browser that the JavaScript code is all done and it can go back to HTML code.

2. A **JavaScript URL** is a special kind of URL that starts the execution of JavaScript code rather than branching off to a new page.

JavaScript URL	Lines	What it does
<A HREF="JavaScript:void(0)">	38, 43, 48, 60, 64, 68	Tells your browser to do nothing when the link is clicked.
<A HREF="JavaScript:window.history.go(-1)">	71	Tells your browser, when the link is clicked, to access its History List and to load the last page in this list.

3. **Event Handlers:** Event Handlers run JavaScript code to handle events that occur in your browser such as moving your mouse over a link or off of a link. Event Handlers always begin with the word "on".

Event Handler	Line	Event that triggers it:	What it does:
onClick	60,64,68	mouse clicks over link	Executes JavaScript code called popUp(whichFlower).
onMouseOver	39,44,49	mouse moves over link	Creates a pop-up window and loads an HTML or JPG file into it.
onMouseOut	40,45,50	mouse moves away from link	Deletes pop-up window.

Comments

<!-- (on line 6) and **//-->** (on line 21) are special HTML comments to hide JavaScript from old browsers (Netscape 1.x and Microsoft IE 2.x) that cannot interpret it.

Comments are notes you put in your code that will be ignored by your browser. Comments help you remember things about your code. // on lines 12,13,17, and 18 tells your browser to ignore everything that follows it on that line.

For longer comments that require more than one line, you can embed them between the symbols /* and ***/.** /* (on line 7) tells your browser to ignore everything until */ (on line 10) is encountered.

Here's an example of a JavaScript program.

**Scripting In JavaScript*

```
<HTML>
<HEAD>
<TITLE>Wildflowers</TITLE>

<SCRIPT Language="JavaScript">
<!--//
   /*
     This JavaScript function pops up a window and displays a flower
     in it whenever you click on the name of the flower.
   */
function popUp(whichFlower) {
    //Check which flower was clicked
    //
    if (whichFlower == "chicory") file = "chicory.jpg";
    else if (whichFlower == "foxglove") file = "foxglove.jpg";
    else file = "eveningprimrose.jpg";
    //
    //Open a window and display the flower
    newWindow = window.open(file,'','width=300,height=397');
    newWindow.focus();
}
//-->
</SCRIPT>
</HEAD>

<BODY link="green" alink="green" vlink="green">

<center>
<TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR><TD width=350>
<FONT size=7 color="pink"><CENTER><B>Wildflowers</B></CENTER></FONT>

<FONT size=5 color="green"><B>Welcome to Wildflowers.<P>Wildflowers can be found
in
almost any habitat - wetlands, fields, woods. They can even grow in the cracks
of a 35 road or sidewalk.

Some wildflower plants bloom every year (
<A HREF="JavaScript:void(0)"
onMouseOver="popup=window.open('perennial.html','','width=300,height=200')"
onMouseOut="popup.close()">perennials</A>),

some bloom every other year (
<A HREF="JavaScript:void(0)"
onMouseOver="popup=window.open('biennial.html','','width=300,height=200')"
onMouseOut="popup.close()">biennials</A>),

and some die after a year (
<A HREF="JavaScript:void(0)"
onMouseOver="popup=window.open('annual.html','','width=300,height=200')"
onMouseOut="popup.close()">annuals</A>).</B></FONT>
</TD>

```

```
53 <TD width=225><img src="lathyrus.jpg" width=225 height=315></TD></TR></TABLE>
54
55 <FONT SIZE=5 COLOR="green"><B>Here are some wildflowers we have in West
56 Virginia.</B></FONT>
57
58 <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0>
59 <TR><TD WIDTH=150>
60 <A HREF="JavaScript:void(0)" onClick="popUp('chicory')">
61 <FONT SIZE=5 COLOR="pink"><B>Chicory</A></B></FONT></TD>
62
63 <TD WIDTH=150>
64 <A HREF="JavaScript:void(0)" onClick="popUp('foxglove')">
65 <FONT SIZE=5 COLOR="pink"><B>Foxglove</B></FONT></A></TD>
66
67 <TD WIDTH=200>
68 <A HREF="JavaScript:void(0)" onClick="popUp('primrose')">
69 <FONT SIZE=5 COLOR="pink"><B>Evening Primrose</B></FONT></A></TD></TR></TABLE>
70
71 <A HREF="JavaScript:window.history.go(-1);"><IMG SRC="back.gif" border=0></A>
72 </CENTER>
73
74 </BODY>
75 </HTML>
```

***Scripting In JavaScript.** View this example online.

Tools You Need

Only a text editor (such as Notepad or WordPad) and a JavaScript-capable Web browser are required to develop with JavaScript. Netscape Navigator 2.0 or later or Microsoft Internet Explorer 3.0 or later can interpret JavaScript code. You don't have to be connected to the Internet to write and test your programs. No special browser plug-in is required to display JavaScript. Such ease and accessibility encourages you to try programming with it.

In order to write programs in JavaScript it is necessary to have a working knowledge of HTML. Today's workshop will assume that you are familiar with:

- switching back and forth between an HTML editor and a web browser
- the difference between the HEAD section and the BODY section of a web page
- how to create a hyperlink to another page or site on a web page
- changing the background color of a web page
- HTML Form objects: buttons, text boxes, select groups.

Copying JavaScript Code

When you start writing JavaScript code, it is a good idea to start with code that someone else has written and that performs the same or similar functionality you need. This provides a framework for you to work with and you have only to customize it for your particular needs.

There's nothing JavaScript programmers like to do more than share their work. You will be amazed at the free material floating around. Even absolute beginners can cut and paste JavaScript code that's available on the Web and add it to their own pages. You know enough about JavaScript already to do this. Here's what you do.

1. View the document source of the page you like. Locate the JavaScript code embedded inside the HTML

Find all JavaScript inside <SCRIPT> tags.
Find all Event Handlers.
Find all JavaScript URLs.

2. Copy all the JavaScript code into your HTML document.
3. Copy any Copyright information into your JavaScript code.
4. Customize the JavaScript code to suit your needs.

****Copying JavaScript Code***. Find all JavaScript code in this example. Do the in-class activity.

Document Object Model (DOM)

JavaScript is an object-based language. Objects are its basic building blocks. The majority of what you do in JavaScript involves interacting with objects.

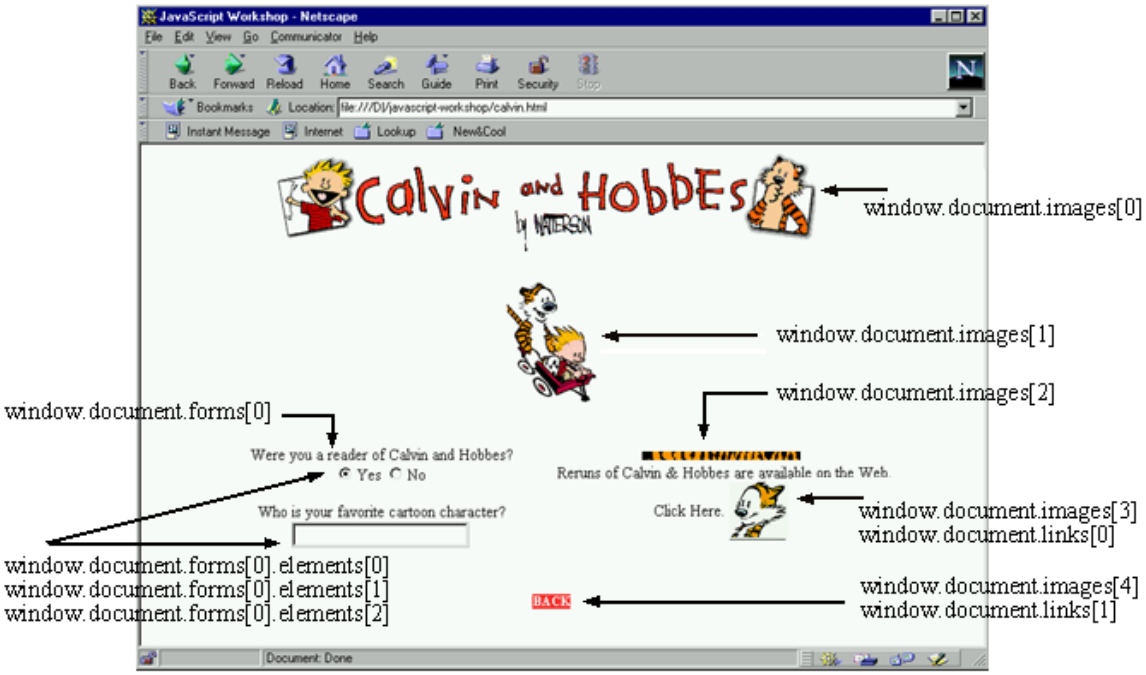
Objects can be described as packages of data. They have

- a set of characteristics (**properties**) that describe their appearance,
- a set of things they can do (**methods**) that describe how they behave,
- a set of actions (**events)** that they can listen for, detect, and respond to. A mouse click is an event.

Your browser and all the images, links, buttons and forms displayed on a Web page are objects. The Document Object Model (DOM) defines and describes these objects, their properties and methods.

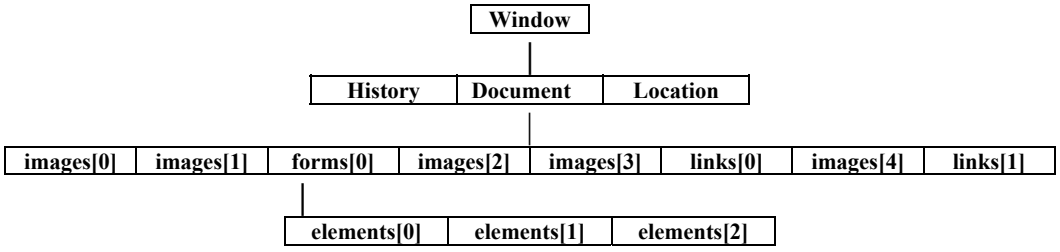
***JavaScript Document Object Model.** Here's a list of objects in the DOM, along with their names and what properties and methods they have.

Here is a simple example of a Web page to illustrate how the DOM works.



Objects

Objects are the things you interact with in a browser. They have a hierarchical relationship. This diagram shows the relationship of the objects on the sample page above.

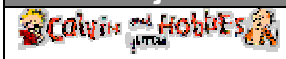
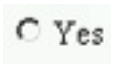
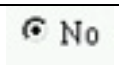


The Window Object is the top-level object in the DOM. (The Window Object is the browser.) The Document Object is a subset of the Window Object. (The Document Object is the page that's displayed in the browser.) JavaScript breaks down the Document Object into more objects. The images, links, and forms displayed on the page are also objects and are subsets of the Document Object.

****JavaScript Objects.*** Summary of JavaScript Objects.

Here's a list of some of the objects on this page and the HTML code that created them. Listed also are two names by which each object is known by JavaScript:

1. The name assigned by JavaScript. JavaScript uses ***Dot Notation*** to refer to objects. It includes the names of all objects along the path from the window object down to the object with a dot separating them.
2. A more descriptive name that I assigned to each object when I created it in HTML.

Object	HTML Code	JavaScript Name
	<IMG SRC="calvin.gif" BORDER=0 **NAME="header"**>	window.document.images[0] window.document.header
	<IMG SRC="wagon.gif" BORDER=0 **NAME="wagon"**>	window.document.images[1] window.document.wagon
form[0]	<FORM **NAME="comics"**>	window.document.forms[0] window.document.comics
	<INPUT TYPE="radio" **NAME="reader"** VALUE="Yes" onClick=Reader()>Yes	window.document.forms[0].elements[0] window.document.comics.reader[0]
	<INPUT TYPE="radio" **NAME="reader"** VALUE="No" onClick=Reader()>No	window.document.forms[0].elements[1] window.document.comics.reader[1]
	<INPUT TYPE="text" **NAME="favorite"** SIZE=20 onChange="FavoriteChar()"> </FORM>	window.document.forms[0].elements[2] window.document.comics.favorite
	<IMG SRC="tail.jpg" BORDER=0 **NAME="tail"**>	window.document.images[2] window.document.tail
link[0]	<A HREF="http://www.uexpress.com/ups/comics/ch" onClick="HappyReading()";>	window.document.link[0]
	<IMG SRC="hobbes2.jpg" **NAME="hobbes"** border=0> </A>	window.document.images[3] window.document.hobbes
link[1]	<A HREF="JavaScript:window.location='../index.html';">	window.document.link[1]
	<IMG SRC="back.jpg" **NAME="back"** border=0> </A>	window.document.images[4] window.document.back

Note: 'window' may be omitted from the JavaScript name. For example, you can refer to the BACK button as, **document.images[4]**, instead of the longer name, **window.document.images[4**].

If your code refers to objects located in the same window as it resides, then you can omit 'window' from the name of the objects. If objects are located in other windows, then you must give each window a name and use that name in all Dot Notation references to objects. For a full, detailed description of the DOM, refer to the Netscape Documentation Library: developer.netscape.com/docs/manuals/index.html?content=javascript.html.

Properties

Properties describe how objects look. You can change the properties of an object by using a notation called Dot Notation. You simply assign a new value to the property you wish to change: **objectName.propertyName = value**

Using the sample page as an example, move your mouse over the graphic with Calvin and Hobbes riding in the wagon. It changes. Now move your mouse off of it. It changes back. The following code implements this image swap.

```
<A HREF="#" onMouseover="document.wagon.src = imageOn.src;"
        onMouseout="document.wagon.src = imageOff.src;">
<IMG SRC="./wagon.gif" BORDER=0 NAME="wagon"></A>
```

Not all objects have the same properties. For a list of some JavaScript objects along with the properties associated with them, view the Summary of JavaScript Objects.

****JavaScript Objects.*** Summary of JavaScript objects and their properties.

Methods

Methods perform actions on objects. Using the sample page as an example, click on the Yes radio button and watch for a small popup window with a picture of Calvin. The following code creates and manages this popup window. All methods used in this code appear in bold type.

```
function Reader()
{
      popup = window.open("","","width=100,height=125");
      popup.document.open();
      popup.focus();

popup.document.write("<HTML><HEAD><TITLE>Calvin</TITLE></HEAD>
         <BODY><img src='calvin.jpg'><BR>
         <FORM><input type='button' value='Close'
onClick='window.close()'></FORM>
        </BODY></HTML>")
      popup.document.close();
}
```

Not all objects have the same methods. For a list of many of the JavaScript objects along with the methods associated with them, view the Summary of JavaScript Objects.

****JavaScript Objects.*** Summary of JavaScript objects and their methods.

Events

JavaScript-enabled Web pages are typically event-driven. Events are actions that occur on the Web page. Generally speaking, they occur when:

Your browser does something such as loading or unloading a page.
You do something like clicking a button, moving your mouse over a link, or moving your cursor into or out of a form element.

It is these events which cause JavaScript code to spring into action. Each JavaScript event has a corresponding event handler that is charged with the responsibility of automatically responding to it when it occurs.

Event Handlers

Event Handlers execute JavaScript code to respond to events whenever they occur. They are what makes JavaScript so useful in creating interactive Web sites.

This table lists the names of event handlers, their corresponding events and objects. Notice that event handlers begin with the word "on".

Event Handler	Event	Object
onClick	A user clicks on an object.	Button, Checkbox, Radio, Link, Reset, Submit, Area
onChange	The value (usually the text) of an object changes.	Select, Text, Textarea
onSelect	A user selects a textbox in a form.	Text, Textarea
onFocus	A user moves a mouse into an object.	All Form Elements, Window
onBlur	A user moves a mouse away from an object.	All Form Elements,Window
onMouseOver	The mouse moves over an object.	Link, Area
onMouseOut	The mouse moves off an object.	Link, Area
onSubmit	A form is submitted.	Form
onReset	A form is reset.	Form
onLoad	A document or an image has finished loading.	Window
onUnload	A document is closed.	Window
onAbort	A user stops loading of an image.	Image
onError	A document or image cannot load.	Window, Image

****JavaScript Objects*.** Summary of JavaScript objects and their EventHandlers.

Using the sample page as an example, the event handlers onMouseOver and onMouseOut process the events that occur as you move your mouse over and off of the image of Calvin and Hobbes in the wagon. Event handlers can be placed with the HTML code that creates the object on which they act.

```
<A HREF="#"  onMouseover="document.wagon.src = imageOn.src;"
        onMouseout="document.wagon.src = imageOff.src;">
<IMG SRC="./wagon.gif" BORDER=0 NAME="wagon"></A>
```

Browser Incompatibilities

Netscape and Microsoft browsers sometimes treat the same JavaScript code in different ways. That's because the JavaScript contained in their browsers is not the same. When you use JavaScript code in your Web pages, you should view your pages in both browsers to be sure your JavaScript code works equally well in both browsers.

As a Web developer your problems arise because JavaScript has undergone several version changes since its release and there are numerous versions of the browser being used today. Unfortunately, this can often lead to confusion and incompatibilities. Here's a list of some popular browsers used today and the JavaScript version they support.

Browser	JavaScript Version	Browser	JavaScript Version
Netscape 2.0	1.0	MS Internet Explorer 3.0	1.0
Netscape 3.0	1.1	MS Internet Explorer 4.0	1.2
Netscape 4.0 – 4.05	1.2		
Netscape 4.06 – 4.7	1.3		
Netscape 6.0	1.5		

Here are some Web sites that discuss browser compatibility issues:

- developer.netscape.com/viewsource/index_frame.html?content=goodman_roadmap/goodman_roadmap.html
- www.webreference.com/js/column11

Debugging in JavaScript

There are three general types of errors:

Load-time errors. caught by JavaScript as your browser loads a script. JavaScript displays a warning box telling you the problem and, most of the time, shows you the actual text and line number in which the error occurred.

Run-time errors. occur when the script is actually running. JavaScript displays an alert box telling you the nature of the error, along with a line number (which isn't always accurate) of where it occurred.

Logic errors. when your script runs but does something different than you intended.

When an error message denotes a line number where the error occurred, that line is counted down from the top of the HTML document, not the top of the JavaScript. You should count all the lines, even the blank ones.

If you are using Netscape, you can type **javascript:** in the location box to view error messages in more detail.

Links to JavaScript Resources on the Web

There is no shortage of materials on the Web related to JavaScript. The Web is a great place to learn more about JavaScript. These sites offer tips, tricks, tutorials, and sample code.

Documentation

Netscape JavaScript Guide
developer.netscape.com/docs/manuals/communicator/jsguide4/index.htm

Netscape JavaScript Documentation Library
developer.netscape.com/docs/manuals/index.html?content=javascript.html

Microsoft Jscript Documentation Library
msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsoriJScript.asp

JavaScript Examples, Articles, and Tutorials

javascript.internet.com/

www.javascripts.com

www.mountaindragon.com/javascript/ <- a discussion list

www.webreference.com/javascript/

www.webcoder.com/index2.shtml

www.webmonkey.com

www.javaworld.com/javaworld/topicalindex/jw-ti-javascript.html

www.jsworld.com/

All the usual search engines can come up with additional sites when you search on the keyword "JavaScript." For instance, Yahoo has a nice collection listed under: **Computers > Internet > Programming Languages > JavaScript.**

***JavaScript References On Web.** View these links online.