Using JavaScript to build a psychology practice exam

JESSE S. HARRIOTT
DePaul University, Chicago, Illinois

The present paper introduces JavaScript scripting in an applied context for social scientists. First, basic JavaScript features are presented. Second, the paper describes a JavaScript application that permits students of psychology to take up to three practice exams. The students are presented with a Web page that poses three sets of five multiple-choice exam questions (in random order). The student is asked to answer each question and is given immediate feedback about his/her performance. In addition, a running total of the correct answers is updated each time the student answers a question, giving the student feedback on overall performance. The application is easily customizable to allow instructors to add, change, or delete questions.

Today the Web is an information exchange medium for approximately 9 million people (Meeker & DePuy, 1996). It is now possible to share text, images, sound, and video on the Web, and the Web is becoming more interactive. Sun Microsystem's introduction of the object-oriented programming language Java, and its counterpart JavaScript, has led to an increasing number of Web authors adding interactivity to their Web pages. With JavaScript, one can add interactive features to Web pages such as the following: user selection of color for Web pages; calculators; dynamic output based on user environment or responses; form verification; and the building of custom URLs. In addition, all of JavaScript's processing is done by the client's computer, which eases bandwidth requirements, decreases server load, and reduces security risks to the server.

JavaScript has much to offer psychologists in the way of greater interactivity on Web pages used for on-line research studies, teaching tools, on-line testing of students, dissemination of research and so forth. Furthermore, JavaScript has a lower learning curve than do other programming languages, which makes it excellent for first-time programmers. The present paper offers an introduction to JavaScript scripting in the context of creating a simple psychology department home page. First, a basic JavaScript script, easily incorporated into an HTML document, is presented. Second, the paper provides an introduction to JavaScript dialog boxes and form processing. Last, a JavaScript application is presented with which teachers of psychology can give up to three practice exams to their students. All of the source code for the JavaScript application is available for anonymous FTP at <ftp://ftp.depaul.edu> in the "/pub" directory. This application should be of special interest to teachers of psychology

who want to allow their students a convenient, anonymous, easy way to test themselves and their progress on the course material.

JavaScript Objects

At the present time, JavaScript is only supported by Netscape Navigator (Version 2.0 or later). JavaScript is a scripting language, which means it requires less effort than a programming language such as Java or C++. JavaScript is object-based, with several sets of built-in objects that provide information about the currently loaded Web page.

In this paper, I assume that the reader has knowledge of HTML. Furthermore, the JavaScript and HTML in Example 1 will be used throughout the paper to illustrate JavaScript concepts. When viewed through a Netscape browser, Example 1 produces a psychology department home page that has three main features: (1) it greets the user via a pop-up box, (2) it personalizes the home page with the user's name, and (3) it gives the user the opportunity to submit comments about psychology classes. The meaning of each JavaScript command will be explained throughout the paper. The code for Example 1 is shown in Listing 1.

The easiest way to incorporate JavaScript into a Web page is to make it an integral part of the HTML document by using the `<SCRIPT>` tag, as seen in Example 1. The line with the `<SCRIPT>` tag tells the browser to expect a script of the `LANGUAGE="JavaScript"` to follow. The next line is used to tell browsers that do not support JavaScript to hide the upcoming script from the user. This allows the page to load without difficulty and prevents browsers that do not support JavaScript from displaying the script. The line `//STOP HIDING FROM OTHER BROWSERS-->` tells the browser to stop hiding text, and the `</SCRIPT>` tag tells the browser that the script has ended.

Another way to include JavaScript scripts in an HTML document is to use the `SRC` attribute of the `<SCRIPT>`

The author thanks Ching-Fan Sheu for discussions concerning this paper. Correspondence should be addressed to J. S. Harriott, Department of Psychology, DePaul University, 2219 N. Kenmore Ave., Chicago, IL 60614 (e-mail: jharriot@condor.depaul.edu).

Listing 1

```

<HTML>
<HEAD>
<TITLE> JavaScript Psychology Department Page: Example #1</title>
<SCRIPT LANGUAGE="JavaScript">
<!-- HIDE FROM OTHER BROWSERS
function checkClass(field) {

    // Check if user has entered information in the field
    if (field.value == "") {
        // User has not entered anything
        alert("You must direct your comments to a class.");
    }
}

alert("The Generic Department of Psychology Welcomes You!")

var name=prompt("Please tell us your name:","Name");
document.write("<CENTER><H1>Generic University</H1></CENTER>")
document.write("<CENTER><B><H2>Department of
Psychology</H2></B></CENTER>")
document.write("<HR><B><H4>Greetings,
"+name+"<BR>Welcome!</H4></B><HR>")

//STOP HIDING FROM OTHER BROWSERS --->
</SCRIPT>

</Head>
<BODY>
<H2>Class Opinion Page</H2><HR></CENTER>
<B>Please submit your constructive comments (anonymously) about
the psychology classes offered this semester at Generic
University</B><br>
<FORM METHOD = POST ACTION ="/cgi-bin/comments.pl"
<B>Class</B>
<INPUT TYPE=text NAME="class" SIZE=45><br>
<B>Comments</B>
<TEXTAREA NAME="comments" ROWS="3" COL="45"
onFocus="checkClass(form.elements[0]);"></TEXTAREA><BR>
<INPUT TYPE="SUBMIT" VALUE="Submit">
</FORM>
</BODY>
</HTML>

```

tag. Using this attribute allows the script to be saved as a separate file that is invoked at load time by the browser. The exact way to specify a file containing a JavaScript script is to replace the `<SCRIPT>` tag seen in Example 1 with `<SCRIPT LANGUAGE="JavaScript" SRC="http://www.you.edu/JavaScript.js">`, where `JavaScript.js` is the name of the script to be loaded by the browser. This method clearly has advantages when the script being written is long or needs to be changed frequently. A disadvantage is the additional server request, which may result in longer load time in a slow server or Internet connection.

JavaScript is an object-based scripting language; it includes many built-in objects that allow the programmer to work with elements of the currently loaded HTML document. JavaScript also includes objects for complex mathematical calculations as well as client processing of forms. *Objects* are where a JavaScript script can store information about the client browser and its activities. In addition to these built-in objects, JavaScript allows for the programmer to create his/her own objects.

JavaScript includes built-in methods under each of these objects. *Methods* are ways in which JavaScript can interact with the client browser, such as opening a new window, changing color preferences, or writing text to the screen. Objects only allow for information to be obtained about the client browser, whereas methods allow the client browser session to be changed in one way or more. For a more detailed description of all built-in JavaScript objects, please consult one of the many books on JavaScript, or visit Netscape's (1996) JavaScript documentation site at <http://home.netscape.com/eng/mozilla/Gold/handbook/javascript/index.html>.

The *document* object is a built-in JavaScript object that provides information contained within an HTML document such as forms, links, anchors, the current URL, colors, and the title. For example, in a JavaScript script, the document object can be used to store information such as the color of the current browser window, the current URL, and the title of the document. Example 1 shows the document method `document.write()`, which takes a text string as its argument. This method

permits the browser to display the text string in the current window of Netscape. The text can be formatted by placing any HTML tag within the quotation marks of the text string. In Example 1, the `<CENTER>` before the text in the `document.write()` command centers the text in the browser window. The `document.write()` method can also be used to display text to the browser screen in response to a user event.

Dialog Boxes

JavaScript provides the ability to generate output in small dialog boxes. The dialog box is independent of the Web page and does not affect its appearance. Dialog boxes pop up under predefined circumstances and can be used to alert the user to information, or to permit interaction with the user. Example 1 contains a command that creates a dialog box that alerts the user that he/she is welcome to the home page. The command `alert("The Generic Department of Psychology Welcomes You!")` creates the alert dialog box seen in Figure 1 containing the message, "The Generic Department of Psychology Welcomes You," as well as an "OK" button. The script waits for the user to press "OK" before continuing to load the home page. Dialog boxes can be helpful in bringing the user's attention to important information. Examples in psychology might include alerting the user to a change in the syllabus, an upcoming assignment due date, or an update on a departmental policy. JavaScript alert boxes include "JavaScript" at the start of the message, in order to distinguish the dialog boxes from those that may originate from the operating system or the

browser. Unfortunately, this is necessary to prevent scripts from tricking users into doing things they do not want to do (Danesh, 1996).

As noted earlier, the `alert()` dialog box contains an "OK" button, which the user must select before the script continues. The "OK" button does allow some control over the timing of events, but it cannot be used to generate dynamic output based on user input. Example 1 also contains script for a dialog box that permits user input. The `prompt()` method is used in the same manner as the `alert()` method, except that it creates a dialog box that allows the user to enter his/her own input instead of just an "OK" button. Just as in `alert()`, the script will not continue until the user enters input and selects "OK." The line `var name=prompt("Please tell us your name:","Name")` prompts the user to enter his/her name and assigns that result to the variable "name." The script then uses the `document.write()` method to display text to the browser window that includes the user's name. Traditional HTML tags can be used within the quotes of the `document.write()` method to format the text. Figure 2 shows the results of entering a name in the prompt dialog box. Input dialog boxes can be used for many purposes, such as to have students answer test questions, to customize on-line examples, or to present lecture notes for a class that a user specifies.

Interactive Forms

Since the introduction of HTML 2.0, forms have been a common feature on many Web pages. HTML forms allow the user to enter information in fields appearing

Figure 1. Alert dialog box created with JavaScript.

Figure 2. Home page created after entering the name "John Doe" in the prompt dialog box.

on the browser screen. The information contained in the form is then submitted by the client browser to the server and processed by a CGI (common gateway interface) program. A CGI program, written in Perl or C, resides in the server and is the standard means for the server to talk to client browsers. However, often the information entered by the user is incorrect, or in the incorrect format. Subsequently, the server is sent information that cannot be processed. JavaScript forms can be preprocessed on the client side to ensure that information is entered correctly before being sent to the server. This ensures that the CGI program will not receive any useless information, and it reduces the load on the server computer.

Example 1 contains a script for a home page that lets psychology students give anonymous feedback to the teachers of psychology classes. The home page has a title, directions, and a two-field form for submitting comments. Users are asked to specify the class to which their comments apply. JavaScript is employed to ensure that the user directs his/her comments to a particular class. If the student tries to write comments without specifying a class, a dialog box pops up to alert the user that he/she must specify a class. Figure 3 shows the result of the JavaScript script.

In order to accomplish this, the function `checkClass()` was created. The function takes *field* as its argument, meaning the text field "class." The function consists of an *if* statement: *if* the "class" text field is empty when the user tries to enter comments, *then* create a dialog box that tells the user, "you must direct your comments to a class." The function `checkClass()` is called when the "comments" text area is clicked on by the user. When this happens, the contents of `form.elements[0]` is returned to the function `checkClass()`. By JavaScript definition, `form.elements` is an array containing an entry for each element in the form (e.g., text fields, checkboxes).

Adding the `[0]` tells the program to take only the first item in that array, which is the text field containing the "class" information offered by the current user.

Psychology Practice Exam

The JavaScript objects, as well as others, were used to create an HTML document that can be customized to give students an on-line psychology practice exam. When viewed in Netscape, the JavaScript application contains four frames: a title frame, a question area frame, an answer area frame, and a test number selector frame. The student is given three sets of 10 multiple-choice exam questions. The student is asked to answer each question and is given immediate feedback about his/her performance. A running total of the correct answers is updated each time the student answers a question, giving the student feedback on overall performance. The psychology practice exam proctor was adopted from JavaScript code contained in *Teach Yourself JavaScript* (Danesh, 1996).

The psychology practice exam contains seven different HTML files. The file "Test.htm" is the page initially loaded by the browser. It produces three frames with a different URL in each: one for the title and instructions, one for the question area, and one for the test selector. The file "Work.htm" contains code for the question area and creates two frames within itself. One frame displays questions, and the second frame displays the score and receives answers from the student. The file "Form.htm" is called by "Work.htm" and contains code for the actual questions and answers for the practice exams. This file contains all of the functions necessary to prompt a student with questions and check his/her answers. The file "Level.htm" is called by "Test.htm" and contains code for the bottom frame that allows a student to select one of three practice exams. The file "Display.htm" is called by "Work.htm" and contains code for the window that

Figure 3. Dialog box generated when user tries to enter comments without specifying a class.

can receive answers from the student and display his/her overall score. The file "Title.htm" is called by "Test.htm" and generates the title "The Psychology Practice Exam" to be displayed in the top frame of the Web page. The last file, "Output.htm," specifies the color to be displayed in output frame. The psychology practice exam can be seen at <http://condor.depaul.edu/~jharriot/test/test.htm>.

To customize the psychology exam for a different test, insert the questions for the test in file "form.htm," using the format presented in the file. Immediately after each question is a number that must correspond to the correct answer in each of one's test questions. For example, if the third choice presented is the correct answer to the question, the number after the question in "form.htm" must be a "3." All seven files are available for anonymous ftp at <ftp.depaul.edu> in the "pub" directory.

Conclusion

The present paper has provided an introduction to JavaScript, as well as an on-line psychology practice exam. Those who wish to learn more about the full benefits and capabilities of JavaScript should consult one of the many books on JavaScript or visit a JavaScript Web site (e.g., www.c2.org/~andreww/javascript/, www.ipst.com/docs.htm, www.txdirecty.net/users/everett/). JavaScript can

offer added student interactivity to the virtual classroom in the form of dialog boxes, form verification, and on-line testing of course material. It is important to keep in mind in writing Web pages that JavaScript is fully functional only on Netscape. Many potential visitors to your site may not use Netscape. Furthermore, Microsoft (1996) has developed its own scripting language, VBScript, with similar features and functions for Internet Explorer. For more information, see their Website at <http://www.microsoft.com/vbscript/us/vbslang/vbstoc.htm>.

REFERENCES

- DANESH, A. (1996). *Teach yourself JavaScript in a week*. Indianapolis, IN: Sams.Net.
- MEEKER, M., & DEPUY, C. (1996). *The Internet report*. New York: Harper Business.
- MICROSOFT (1996). VBScript documentation site. *Microsoft Website* [On-line]. Available: <http://www.microsoft.com/vbscript/us/vbslang/vbstoc.htm>
- NETSCAPE COMMUNICATIONS, INC. (1996). JavaScript documentation site. *Netscape Communication Website* [On-line]. Available: <http://home.netscape.com/eng/mozilla/Gold/handbook/javascript/index.html>

(Manuscript received November 4, 1996;
revision accepted for publication April 4, 1997.)