

40 responses -  
83 % Indicated amount of work required about  
right  
17 % Indicated amount of work required excessive

Comments:

2. The in-class presentations were extremely helpful in understanding the paper discussed on-line.
3. Too little - As far as my university was concerned. I have spoken to the office here and expressed my exasperation with the total lack of instructor involvement in the class.  
Excessive - As far as the list-server was concerned. I think that the organizers need to consider the number of students involved as being a class with no restraints on who speaks or when. Either limiting the number of students or colleges involved, or using a different forum for the discussion would alleviate this problem.
17. Required more work than many of my other classes. This was by no means unmanageable, but you had to devote the time needed to keep pace with the others.
27. I believe this course should be expanded to a two credit course if possible. Ideally, the amount of work required for this course would be worth more than one hour credit.
29. The amount of work required for this course was excessive at times, but well worth it.
34. About right for a three credit course.  
Excessive for the one credit course I took it for, but it was interesting. I believe the course to be very worthwhile and hope it is offered again.

## Programming Methods for the World-Wide Web

**Brian M. Tissue**  
**Department of Chemistry**  
**Virginia Polytechnic Institute and State University**  
**Blacksburg VA, 24061-0212**  
**tissue@vt.edu**

### Introduction

This article provides a basic overview of several common programming methods that are used to create interactive material for the World Wide Web (WWW).<sup>1</sup> Interactive Web pages can be textual, calculational, or graphical; and can serve as on-line versions of tutorials, homework problems, take-home exams, and other educational tools. The largest distinction between the different interactive methods is that some are client-side, in which all processing is done on the client computer, and others are server-side, in which all processing is done on the WWW server. Client-side methods include client-side image maps, JavaScript,<sup>2</sup> and Java. The question-and-answer information for client-side image maps and JavaScripts is contained in the HTML file, and users have access to the information by looking at the HTML source code. Server-side methods include server-side image maps, and scripts or programs called through the Common Gateway Interface (CGI). Server-side methods are most appropriate for delivering interactive exercises that access existing databases of information, or require tracking or grading capabilities. Client-side methods are most appropriate for practice exercises. Client-side methods have a major advantage in delivery speed over server-side methods in that they do not make a new connection and transfer data over the internet for the data processing step. This consideration can be very important for interactive material that requires many repetitive actions. Combinations of client-side and server-side methods are possible. For example, a client-side JavaScript can check the format of data entry before it is submitted to a server-side program for evaluation.

### Hyperlinks

Multiple-choice exercises can be produced without any programming by simply using hyperlinks for each answer choice. When a choice is selected, the link retrieves a new HTML page that can indicate if the choice was correct or incorrect, and can also provide an explanation or a return link to try the exercise again. An

extension of this simple hyperlink method is to create a graphical multiple-choice exercise using a clickable image map. Although this approach is simple, it can create a large number of files, requiring careful file management. The major limitation of using only hyperlinks is the inability to create interactive materials that require relational or calculational evaluation of a user's input. Such exercises require a programming method that can be interfaced to the Web.

### Image Maps

Clickable image maps are in-line images that contain active areas, and provide a very simple means of creating interactive graphical components on a Web page. Clicking on an active area calls a specified URL, which can be a hyperlink to a different part of the Web page or to a new HTML file, or a call to a JavaScript function. There are two types of image maps: client-side and server-side, which differ in the placement of the information specifying the active areas. The client-side map information is contained in the same HTML file as the Web page that is being displayed, and the server-side map information is in a map file on the WWW server. Support for client-side image maps was implemented in Netscape Navigator 2.0 and compatible browsers. Client-side maps operate faster than server-side maps because they do not have to reconnect to the WWW server for each click on the image map. Client-side maps are also easier to develop because the files can be tested and edited on a local disk before transferring to a WWW server. Server-side maps have the advantage that the map information is not available to the user, so they can be used as interactive graphical exercises in on-line testing material. Clickable image maps can be created manually by typing in the coordinates of the active areas, but are easily created using one of the free or inexpensive image map programs that are available on the Web, such as MapThis or Mapedit.

- Imagemap Authoring Guide and Tutorial Sites, <http://www.cris.com/~automata/tutorial.shtml>.
- Imagemap Help Page — Instruction, <http://www.hway.net/ihip/>.
- MapThis Home Page, <http://galadriel.ecaetc.ohio-state.edu/tc/mt/> (Win32 version only).
- Mapedit, <http://www.boutell.com/mapedit/> (Windows and UNIX versions only).

### JavaScript

JavaScript is an object-based scripting language that is supported by Netscape Navigator 2.0 and equivalent browsers.<sup>3,4</sup> Internet Explorer supports VBScript and JavaScript, although the degree of language compliance between Internet Explorer and Netscape Navigator is not 100%. Netscape Navigator 2.0x does not print a Web page generated by JavaScript commands, but Netscape 3.0 and compatible browsers will print script-generated pages. JavaScript is like other

scripting languages in that it is fairly easy to use, and allows Web authors to write programs without a full-featured compiled language. The JavaScript code is contained in the HTML document, usually in the header section between `<SCRIPT>...</SCRIPT>` tags. The JavaScript is interpreted by the user's browser when a new HTML file is loaded, and any JavaScript statements that write to the page are executed before the HTML code in the body of the document is interpreted and displayed. JavaScripts can recognize and respond to a number of user events such as mouse clicks, form input, and page navigation. Form elements such as radio buttons and text boxes allow users to enter data, such as selecting choices or typing text, that can be evaluated by a JavaScript function, which then returns a response. (The form elements are the same as used to submit information to a WWW server for a CGI program to evaluate.) The disadvantage of using JavaScript alone is that no results from the user's actions can be saved to the server.

- JavaScript Authoring Guide, Netscape Communications Corp., <http://www.netscape.com/eng/mozilla/Gold/handbook/javascript/index.html>.
- JavaScript Index, Andrew Wooldridge, <http://www.c2.org/~andreww/javascript/>.
- The Complete Idiot's Guide to JavaScript, Aaron Weiss & Scott J. Walter, <http://www.winternet.com/~sjwalter/javascript/index.html>.
- Interactive Exercises for On-Line Education, <http://www.chem.vt.edu/chem-ed/CHP/scripts/> (JavaScript and Perl examples).

### Common Gateway Interface

The Common Gateway Interface (CGI) is a standard protocol that allows information to be passed between a browser program running on a client computer and an application program running on a WWW server. When a browser requests a \*.cgi file, the server executes the script or program that is contained in that file. The CGI interface also passes any data that was submitted from the client to the program. The program evaluates the user input and returns a response. CGI can interface to any language that can be executed on a WWW server, such as C/C++, Fortran, TCL (Tool Command Language), PERL (commonly called Perl, although it is an acronym for Practical Extraction and Report Language), VisualBasic, HyperCard, and AppleScript. Non-compiled languages, such as PERL, are usually preferred due to their ease of modification, maintenance, and debugging (PERL is also free). CGI programs are executed on the WWW server using the resources of the server computer. Processor-intensive applications or the ability to serve many clients simultaneously can create a heavy load for a WWW server, requiring a powerful computer. The advantage of the CGI method is the ability to use sophisticated program-

ming languages, and to have access to the server's hard disk to use existing databases or log user responses.

- The Common Gateway Interface, <http://hoohoo.ncsa.uiuc.edu/cgi/>.
- The Perl Language Home Page, <http://perl.com/perl/index.html>.

## Java

Java was developed by Sun Microsystems, Inc. to create a simple object-oriented and platform-independent programming language to distribute application programs over the Web.<sup>6</sup> Java programs are called applets, and are included in HTML code. To create a Java applet, you write a Java program and compile it into a file of Java bytecodes, and include a reference to the applet using `<APPLET>...</APPLET>` tags in an HTML file. When a Java-enabled browser encounters the applet tags, the browser downloads the specified Java applet, interprets the Java object code, and runs it in a window on the browser screen. Applets are stand-alone programs that can accomplish a broad range of functions such as animation, calculations, graphing, etc., and could find wide utility in the creation of interactive learning material. However, Java is a highly syntax-dependent language, requiring some programming skill by the applet author, and the language is still at an early stage of development.<sup>7</sup> Microsoft is developing a similar technology called ActiveX. Currently Internet Explorer 3.0 supports Java and ActiveX but Netscape Navigator 3.0 only supports Java.

- Java - Programming for the Internet, <http://java.sun.com/>.
- All About Java(tm), <http://java.sun.com/aboutJava/index.html>.
- Java(tm)/HotJava(tm) Frequently Asked Questions — Index, <http://java.sun.com/faqIndex.html>.
- VBScript and ActiveX FAQ, <http://www.rollins-assoc.com/vbsfaq.html>.

## Visualization Methods

There are several methods for viewing and manipulating 3-D structures on the Web. Simple 2-D and 3-D visualization can be created with drawing programs and converted to simple or animated images. The following methods include some level of interactivity, in that the user can manipulate the 3-D object, or use the visualized object in an interactive exercise (e.g. hyperlinks in VRML or QTVR). Various representations of molecules; ball-and-stick, space-filling, etc.; in Brookhaven protein data bank (\*.pdb) and other file formats can be displayed and manipulated with the RasMol program. RasMol can also be set up as a helper application for a browser. Alternatively, an evaluation copy of the Chime Plugin for Netscape Navigator displays \*.pdb and \*.xyz files, and provides most of the functionality of the stand-alone RasMol

program. An enhanced version of the RasMol program allows viewing of multiple molecules in the program window, bond rotation, and other features.

- RasMol home page, <http://www.umass.edu/microbio/rasmol> (Win, Mac, and UNIX versions).
- UCB ENHANCED RASMOL, <http://hydrogen.cchem.berkeley.edu:8080/Rasmol/>
- MDL Home Page (Chemscape Chime plug-in), <http://www.mdli.com/>

Virtual Reality Modeling Language (VRML) allows visualization and manipulation of 3-D objects in a window on a Web page. The user can change the position from which the object is viewed, including zooming in and out. VRML objects can include hyperlinks making it possible to use them as interactive 3-D exercises. Apple has developed similar panning and linking capability for QuickTime movies called QuickTime VR. VRML files have a \*.wrl extension and QuickTime VR files have the same \*.mov extension as regular QuickTime movies. VRML appears to be used mostly for 3D renderings of objects, and QuickTime VR for panoramic scenes created from video.

- VRML Architecture Group, <http://vag.vrml.org/>.
- Welcome to vrml.sgi.com, <http://vrml.sgi.com/>.
- Apple QuickTime VR Home, <http://qtvr.quicktime.apple.com/>.

## Summary

The following table summarizes the methods discussed above and appropriate uses. Java and JavaScript are complementary tools. Java provides sophisticated programs that run in a separate window on the browser screen, and JavaScript provides a simple language that interfaces with Web page elements such as forms, etc. For security reasons, neither Java nor JavaScript can write to a user's hard disk. The CGI interface allows utilization of sophisticated programming languages and access to the server hard disk.

Programming Method	Typical Use
client-side image maps	navigation, graphical practice exercises
server-side image maps	navigation, graphical exercises that logs user answers
CGI	multiple-choice or calculational exercises that access databases or log user answers
JavaScript	multiple-choice or calculational practice exercises, easy to interface form elements and clickable image maps on a Web page
Java	stand-alone presentations or tutorials

This article has attempted to present an overview of Web programming methods. There are a number of other software tools for visualizing and manipulating objects and data that were not discussed in this article. A couple of noticeable ones for education are the Shockwave plug-in that allows Macromedia Director, Authorware, and Freehand files to be viewed from a browser window, and Mathcad 6.0 or Mathbrowser for viewing mathcad worksheets.

- Macromedia: Shockwave Center, <http://www.macromedia.com/shockwave/>.
- Mathcad 6.0 Animation Gallery, <http://www.mathsoft.com/60dir/animatio.htm>.
- Mathbrowser Home Page, <http://www.mathsoft.com/browser/index.html>.

#### References

Note: all URLs listed in this article were verified 9/16/96.

- <sup>1</sup> A good starting point to find programming and Web tools is: Yahoo! - Computers and Internet:Internet:World Wide Web, [http://www.yahoo.com/Computers\\_and\\_Internet/Internet/World\\_Wide\\_Web/](http://www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/).
- <sup>2</sup> Not to be confused with a server-side version of JavaScript, which is under development.
- <sup>3</sup> Mark. C. Reynolds and Andrew Wooldridge, Special Edition Using JavaScript (Que, Indianapolis, IN, 1996); ISBN: 0-7897-0789-6.
- <sup>4</sup> Aaron Weiss and Scott J. Walter, The Complete Idiot's Guide to JavaScript (Que, Indianapolis, IN, 1996); ISBN: 0-7897-0798-5.
- <sup>5</sup> Till, D. Teach Yourself PERL in 21 Days; Sams Publishing: Indianapolis, IN, 1995.
- <sup>6</sup> There are lots of books on Java (and the other topics in this article). O'Reilly sells several Java

books at several levels of expertise; see: Java, <http://www.ora.com/catalog/java.html>.

- <sup>7</sup> JavaScript is also at an early stage of development, but uses less rigorous variable typing and is therefore more forgiving for imperfect program code.

---

#### "Some Reflections On the Intercollegiate On-Line Course: Industrial and Environmental Chemistry"

James M. Beard, Chair of the Organizing Committee  
Catawba College  
Salisbury, NC  
[JBEARD@catawba.edu](mailto:JBEARD@catawba.edu)

In this article I would like to take some time to reflect on the On-Line Chemistry Course which was taught last Spring. I am not going to dwell too much on the mechanical details of the course, as Don Rosenthal did an admirable job of that in his article in the Spring 1996 issue (pp. 16-20). I will express some of my observations about the course.

The organizing committee began its work in January of 1995, a full year before the course began. I would like to thank the committee (George Long, Don Rosenthal, Maria Pacheco, Reed Howald, and Carl Snyder) because I could not have organized this course without them. During this year a considerable amount of work took place. I wasn't sure at the time we could complete the preparations for the course, but we did. In the end there were 20 schools and 96 students which turned out to be about the right number. My thanks again to all those instructors and the authors whose commitment made this course a success.

After all of the thousands of pages of email and the countless hours of studying, guiding and reviewing, what