

EMERGING TECHNOLOGIES

Dynamic Web Page Creation

Robert Godwin-Jones

Virginia Commonwealth University

Contents:

- [Plug-ins and Applets](#)
- [JavaScript](#)
- [Dynamic HTML and Style Sheets](#)
- [Instructional Uses](#)
- [Resource List](#)

While remaining a powerful repository of information, the Web is being transformed into a medium for creating truly interactive learning environments, leading toward a convergence of Internet connectivity with the functionality of traditional multimedia authoring tools like HyperCard, Toolbook, and Authorware. Certainly it is not fully interactive yet, but that is undeniably the trend as manifested in the latest (version 4) Web browsers. "Dynamic HTML," incorporated into the new browsers, joins plug-ins, Web forms, Java applets, and JavaScript as options for Web interactivity.

Plug-ins and Applets

While Web pages are beginning to behave more like interactive applications, traditional authoring tools are themselves becoming Internet-savvy, primarily through the use of "plug-in" versions of players which integrate with Web browsers. The most commonly used plug-in today is Macromedia's "[Shockwave](#)," used to Web-enable such applications as Director, Authorware, and Flash. "Shocked" Web pages can be very interactive and provide a visually appealing means of interacting with users (as in some sample [ESL exercises](#) from Jim Duber).

Plug-ins are easy to use -- they just need to be downloaded and installed. Some come bundled with Netscape and Microsoft's browsers, which simplifies considerably the installation process (and gives developers the confidence that most users will actually have the plug-in installed). Plug-ins are not themselves tools for creating Web interactivity. They simply provide a means to put on the Web applications created by traditional authoring systems. The extent to which plugged-in programs are integrated into Web pages or are Internet-aware (so that they can exchange information with a Web server, for example) varies considerably.

The big name in terms of Web interactivity is [Java](#), a programming language about which there has been a huge media buzz for the past two years or so. Using Java makes it possible to create small applications (applets) which, like most plug-ins, are seamlessly integrated into a Web page. In fact, Java can be used to create stand-alone applications as well. Its appeal is its portability to different computing systems (not just standard desktop operating systems), its built-in Internet connectivity, and its relative ease of use (for programmers!). Java is being widely used in disciplines such as [physics](#) and [statistics](#) to create real-time simulations on the Web.

The power of Java comes at the cost of having to learn a full-fledged programming language, not something to be recommended for the casual Web author. There are, however, ways to take advantage of Java without having to program. There are collections of Java applets which allow for use of applets to do such things as create scrolling banners or animated buttons. Such applets can be used to spice up a Web page, but they don't really tap into the enormous potential of Java for creating a dynamic learning environment. Fortunately, there are some ways now developing for harnessing the full power of Java. There are "drag and drop" Java editors such as the recently released [Java Studio](#) from Sun or Symantec's

Visual Café. Another option is the use of template systems such as [J-BAT](#), a tool under development for creating Java-based quizzes for language learning. Java has great potential for creating sophisticated, network-aware applications, especially real-time collaborative environments and interactive graphic applications. One of the side benefits of Java for language teachers is its capability of using [Unicode](#) to display many different character sets.

JavaScript

Java works essentially the same way as plug-ins. As with plug-in programs, Java applets are small applications which are downloaded separately from the accompanying Web page. One of the more recent ways to create interactivity provides a means for doing so while being incorporated into normal Web pages. This is possible through the use of "client-side" functionality built into the latest round of Web browsers. The most familiar approach is [JavaScript](#). JavaScript is a scripting language created by Netscape in 1995, which has been incorporated into their Web browsers since then (versions 2, 3, 4, and Communicator). It shares some basic syntax with Java, but JavaScript is not a stand-alone programming language -- it functions only within a Web browser. It is not as powerful as Java, but it is much easier to learn and to use.

With each new version of their Web browser, Netscape has increased the functionality of JavaScript to the point that it can be used today to create dynamic interactions in a Web page. JavaScript can be used to generate actions in response to mouse clicks or even to cursor movements. Those actions can range from putting a message in the "status bar" (at the bottom of the browser window), to changing an image on the page, opening a new window, or adding text to a document in a frame.

One of the most useful attributes of JavaScript is the ability to work with form elements on a page such as radio buttons, checkboxes, or text fields. It used to be that the only way to get at the data from a Web form was to use [CGI](#) (Common Gateway Interface). This involves writing a program (most commonly in [Perl](#)) which resides on the server and goes into action once a user has submitted a Web form. The CGI script takes the data from the form and typically analyzes it, stores it (in a file or in a database), and sends feedback to the user in the form of a Web page. It is particularly useful for creating collaborative learning environments such as discussion forums.

With JavaScript it is possible to provide feedback based on information entered into a Web form without having to send the data to a Web server. All the interaction is local -- within the user's Web browser -- and therefore can be significantly faster. There is also quite a bit more flexibility in how to provide feedback; the form in its entirety does not need to be submitted (as is the case with CGI ([examples](#))). A quiz, for example, can be analyzed question by question rather than submitted all at once for analysis, thus providing a learning experience in addition to formal assessment (see examples in [ESL](#), [Spanish](#), [French](#), or [German](#)). However, if the answers need to be recorded, then that data must be returned for central storage to the Web server through a CGI script (or a Java applet).

JavaScript has grown to be quite popular on the Web and is being used widely in a great variety of ways. Indeed, it has proven to be a bit more useful than many perhaps originally expected. One of the principal reasons is how it is incorporated into a Web page. JavaScript is not compiled (translated into machine code) like Java; it appears as text alongside the HTML used to generate the Web page. That means that all that is needed to create JavaScript is a text editor. It also means that anyone viewing a page containing JavaScript can see (and borrow) the code by viewing the source of the page -- a process easily done within any Web browser. Newer HTML editors such as [FrontPage](#) or [PageSpinner](#) make it even easier to use JavaScript by making some of its functionality available in pull-down menus which automatically add the necessary code to your Web page. Some editors, like PageSpinner, let you save your JavaScript (and HTML) in 8-bit characters (using the international standard character set [ISO 8859-1](#)), which is crucial for language learning use.

One issue of concern for users of JavaScript has been compatibility. Originally one had to use Netscape Navigator 2 to be able to run JavaScript. Soon, however, Microsoft created "JScript" and incorporated it into their Internet Explorer browser. However, since JScript was reverse-engineered and not derived from licensing JavaScript, it did not (and does not) provide complete compatibility with JavaScript. Fortunately, in a rare display of cooperation, both Netscape and Microsoft have recently agreed to a common scripting standard based on the current version of JavaScript, with some changes. The official standard, [ECMAScript](#), has been endorsed by both companies and will be incorporated into their Web browsers. However, compatibility is still an issue if one is concerned about other types of Web access such as [WebTV](#) (and other set-top TV devices) or hand-held computers (using a browser like [Nethopper](#)). Currently JavaScript is not supported by such browsers, nor is Java.

Dynamic HTML and Style Sheets

JavaScript is a key ingredient in Netscape's implementation of "Dynamic HTML" or [DHTML](#). This is intended to provide a means for changing elements of a Web page "on the fly," that is, after the Web page has been displayed on the screen. Before the current (version 4.0) Web browsers, only certain parts of the page such as images and form elements could be changed after the page had been displayed. With DHTML the goal is to make virtually everything on the Web page able to be changed in reaction to user actions. This includes the possibility of knowing where on the page a user has clicked. By making everything on a Web page both "hot" and changeable, a dynamic environment is being created with functionality similar to what is possible with traditional multimedia authoring systems. Typeface, color, size, visibility, position, and other attributes can all be manipulated on the fly or in response to user actions.

A major addition to the HTML language which enables dynamic access to page elements is Cascading Style Sheets, a standard endorsed by the W3 Consortium as "[CSS1](#)." Style sheets allow Web authors to control globally -- even across an entire Web site -- the appearance of elements on a Web page. This creates a "dynamic object model" ([DOM](#)) allowing user or programmatic access to page elements. In a stylesheet definition, a page element such as a header (with an HTML tag like "<h3>") might be initially defined to display as bold and green, but through a user action or a scripted function could be changed to italics and blue. This kind of interactivity is all local, not requiring any communication with the Web server once the page has been displayed.

Unfortunately, Netscape and Microsoft's implementations of DHTML are different and largely incompatible. Netscape uses a proprietary "layering" system to provide access to all elements of a page. Microsoft supports the newly adopted CSS1 standard, but provides dynamic access to its elements through its proprietary [ActiveX](#) language. Both companies have pledged to support the emerging standards being proposed by the World Wide Web Consortium. For now, the safest way to ensure compatibility is to use the CSS1 standard supported by both browsers without the link to JavaScript or ActiveX. There are editors which will produce DHTML such as [DreamWeaver](#) from Macromedia. The full power of DHTML will be evident when style sheets are tied directly to a common scripting language.

Instructional Uses

What's the benefit of using dynamic Web design in language learning? Many of us have tapped into the rich array of authentic linguistic and cultural Web sites that have sprung up in recent years. To have such a treasure chest of materials in multimedia-capable format available in a networked, cross-platform format is a tremendous teaching and learning resource. Add to this the ability to integrate our own learning materials and create Web pages which transparently link local resources with sites half-way around the world, and we are talking about a very rich and dynamic environment for language learning.

To get started in this area, the best approach is to first explore sites offering examples of dynamic Web pages. A number of these sites offer the possibility of using and customizing the resources illustrated. This means that instructors do not have to re-invent the wheel by authoring from scratch. However, it still requires basic knowledge of how the different options work as well as an understanding of their syntactic structures. Given the pace of development of the Web, it is likely that we will soon be seeing many more authoring options for Java, JavaScript, and DHTML, particularly editors which (like current HTML editors) hide the underlying code. At the same time, to exploit the full power of the medium to provide pedagogical breakthroughs and new paradigms for language learning, it will be necessary to move beyond ready-made, plug-in Web dynamism.

To get a sense of how one might use some of these interactive Web technologies in actual language learning, here are some examples:

CGI (traditional, server-based interactivity)

- [French Grammar Exercises](#) from the University of Texas at Austin
- [Crossword Puzzle in French](#) from Richard Loubejac (Manchester)
- [Spanish Language Exercises](#), both self-correcting (JavaScript) and instructor-corrected (CGI) examples from Juan Ramón de Arana of Ursinus College
- [German 311](#), advanced German class using CGI-based discussion forums, home page creation, and homework submission forms

JavaScript (newer, browser-based interactivity)

- [Ocrat Chinese Pages](#), examples of JavaScript and animated gifs to draw Chinese characters
- [English-Japanese KANJI dictionary](#) from Tatsuja
- [GRIFF](#), intermediate French course from the University of Alberta with extensive use of JavaScript
- [European Capitals Quiz](#) uses graphics combined with JavaScript to quiz on European capitals (by Joe Kraft)
- [Herr Quizmaster](#), JavaScript-based quiz on German numbers (by Kerri Schiller)
- [Russian 201](#), advanced Russian course at Bucknell using JavaScript animated gifs for grammar explanations and self-testing; requires Cyrillic KOI-8 fonts (by Robert Beard)

Java (applets integrated into Web pages)

- [Language Java Applets](#), collection from Developer.com (formerly Gamelan)
- [KanjiFlash](#) Java Applet from Marshall Ramsey (Arizona)
- [DisplayGreek](#) Java applet displays ancient Greek by Bruce Robertson (Toronto)
- [Multilingual WordMatch](#), good example of matching capability in Java (from Sun)

RESOURCE LIST

Java

Information/Documentation/Tutorials

[Java 101](#) basics from ZDNet
[Introduction to Java](#) tutorial from IBM (Windows only)
[Java Tutorial](#) from Sun (all platforms)
[Java World](#) on-line periodical

Collections/Demonstrations

[Applets](#), downloadable Java applets from Sun
[Java Applets Pack](#), applet collection from techweb
[Java Test Generator](#) from Purdue
 ESL, [Spanish](#), [Educational Object Economy](#) from Apple

[Java Applet Rating Service](#) tracks new applets
[Earth Web Chat](#) easily builds Java-powered chat

Software/Tools

[Jamba](#) commercial Java templates
[JDK](#), Java Developer's Kit from Sun
[Emblaze Creator](#) tool for delivering streaming media and presentations through Java

JavaScript

Information/Documentation/Tutorials

[JavaScript for the Total Non-Programmer](#) tutorial
[Creating Language Interactivity on the Internet](#), a series of tutorials on JavaScript and HTML from Juan Ramón de Arana (Ursinus College)
[Bring JavaScript To Life](#)
[Write with JavaScript](#)
[JavaScript Notizen](#), introduction in German
[Curso de JavaScript](#), tutorial in Spanish
[Language Interactive](#), introduction to CGI and JavaScript

Collections/Demonstrations

[Educational Uses of JavaScript](#) from Developer.com
[Java Goodies JavaScript Repository](#)
[JavaScript Tip of the Week](#) from WebReference
[The JavaScript Sourcebook](#) -12-

Software/Tools

[Netscape](#) for JavaScript: Communicator or Navigator 2.0 or higher needed
[Microsoft](#) for JScript: Internet Explorer 2.0 or higher needed

Dynamic HTML and Style Sheets

Information/Documentation/Tutorial

[W3 Consortium](#)
[Web Style Sheets](#)
[Document Object Model](#)
[Positioning HTML Elements with Cascading Style Sheets](#)
[Microsoft](#)
[Microsoft Site Builder: Dynamic HTML](#)
[Dynamic HTML in IE 4.0](#)
[Microsoft Pushes Dynamic HTML](#)
[Netscape](#)
[Netscape Communicator |Dynamic HTML](#)
[Dynamic HTML in Netscape Communicator](#)

Other

[Dynamic HTML Lab](#) from Webreference
[Using Dynamic HTML to Create an Animated Navigation Bar](#)
[The Evolution of Style Sheets](#)
[D.J. Quad's Ultimate Style Sheets Tutorial](#)
[Users Grapple with Pace of Web's Changes Weaving the Web Fantastic](#)
[The Power of the DOM](#)
[InternetView: Rift over HTML's Dynamics](#)

Software/Tools

[Netscape](#) Communicator or Navigator 4.0 or higher needed
[Microsoft](#) Internet Explorer 4.0 or higher needed
[Dreamweaver 1.0](#) DHTML editor from Macromedia [Dynamite](#) DHTML editor from Astound
[iNet Developer](#) DHTML editor from Pictorius

Collections/Demonstrations

[Microsoft Dynamic HTML Gallery](#)
[Netscape Communicator Demonstrations](#)
[Taboca ETC](#), demos of Netscape's implementation of layers and JavaScript
[Dynamic HTML Index](#)
[Yahoo! HTML Sites](#)

All links validated on November 29, 1997