Michael H. Birnbaum

2 An Overview of Major Techniques of Web-Based Research

This chapter presents a summary of major methods useful to Web-based research. It is intended to help researchers decide what techniques they need to learn for their particular type of research.

HyperText Markup Language

The most basic method for formatting, organizing, and linking information on the Web is *hypertext markup language* (HTML). This language contains instructions (commands) called *tags* that tell the client's (i.e., your research participant's) browser how to display information contained in files stored on one or more server(s) on the Web. If you are planning to do research via the World Wide Web (WWW), you need to learn basic HTML. Many free tutorials on the Web teach HTML; there are also books that you can use to teach yourself HTML (Birnbaum, 2001; Fraley, 2004).

HTML files are simple text files that can be created in a simple text editor such as Notepad. HTML files can also be created and edited by Web page development programs like Dreamweaver, a commercial program available from Adobe

(originally created by Macromedia). I advise my students to learn HTML before they even decide whether to purchase such commercial programs. Knowledge of how HTML works helps you to understand what the development programs are doing; it helps you avoid terrible errors that can result in data loss and ruined studies that can occur when an ignorant person uses a program he or she does not understand; and it helps you decide what software, if any, you need. When you understand HTML yourself, you will find it is often easier to work directly in HTML rather than struggle with a "what you see is what you get" type of editor. The problem is that what you don't see may result in a failure to get your data in a usable form, if at all.

HTML allows one to do any study that could have been done with paper and pencil, a slide projector, video recorder, and more. One can insert not only graphics and photos in a survey or experiment (which could have also been done through a paper questionnaire) but also sound and video (which are not easy to implement by paper). For a discussion of media that can be delivered via the WWW, see Krantz (2001). Instructional materials by Krantz for the creation and editing of media are available at http://psych.hanover.edu/NSFATI/. Krantz was one of the first people to conduct research via the Web (for a discussion of validity of Web studies, see Krantz & Dalal, 2002,), and he maintains a site where experiments can be listed as a way to recruit volunteer participants in Web-based research: http://psych.hanover.edu/Research/exponnet.html.

Basic HTML permits use of one question that will either include or skip other items. For example, the question "Do you smoke?" might be used to link to a series of questions about smoking or to skip them, depending on how the participant responds. One can also use HTML to have a person click on his or her birth month to create assignment of participants to conditions. Skipping items and assigning participants to between-subjects conditions are examples of what can be done with hyperlink tags.

Web Forms

A very powerful technique that is built into HTML is known as *Web forms*. For example, this technique allows the client to respond to items in a questionnaire by typing in answers to questions, entering numbers, clicking along rating scales, or choosing from pull-down lists. This method can be used to send data to be saved in a file on a server, from which it can be later downloaded and imported to a statistical package, such as SPSS, or opened in a spreadsheet program, such as Excel. The technique can also be used to append data to the log file of the server or to send data via e-mail to an e-mail account.

Exhibit 2.1 shows a basic Web page containing a form that sends two data to a *Common Gateway Interface* (CGI; a standard for computers to exchange information via the Internet) script. The first datum is the hidden variable whose value is "MyTestAge," which identifies the study. The second datum is the person's age (or whatever he or she typed in the box). In this case, the script is written in the programming language Perl (Practical Extraction and Reporting Language). You can test this example on the supplementary Web site for this chapter.

The <form> tag has an <action> specified. The action in Exhibit 2.1 sends the data to the address of a CGI file written in Perl that saves the data to a Web site, which can be viewed at the supplementary Web site for this chapter.

Your data will be the last information contained in the data file on the server that was written by the CGI. This data file is available to be read on the Web. Normally, one would not allow permission for a data file to be read; however, in this case, the file has been given no protection, so that you can test the example. The script also redirects the user to a "thank you" page.

To send data to the server's log file instead of to this data file, you could change the <form> tag as follows:

<form method=get action=thanks.htm>

EXHIBIT 2.1

A bare bones Web form. This Web form requests just one datum from the participant: age. When the "submit" button is clicked, this form sends its data to a CGI script called "simple.pl" in the folder called "cgi-bin" in the Web site (see Exhibit 2.2). That script saves the data to the server.

```
<html>
<head>
<title>My First Form</title>
</head>
<body>
Please answer this question.
<form method= "post" action= "http://ati-birnbaum-2008.netfirms.com/
cgi-bin/simple.pl">
<input type= "hidden" name= "00exp" value= "MyTestAge">
<p> What is your age?
<input type= "text" name= "01age" size= "2" maxlength= "3">
<p><input type= "submit" value= "Send the Data">
</form>
</body>
</html>
```

This method (*get*) will append the data to the next URL and to the server's log file, and in this case, it also redirects the participant to a "thank you" page in the same folder as the survey.

CGI Scripts

Exhibit 2.2 presents a simplified version of a Perl script that was originally written by Schmidt to emulate GCI scripts that were created by PolyForm, a program that is no longer supported. This CGI program organizes the data in the order of leading digits on the variable names (e.g., as in Exhibit 2.1). That aspect of the script means that a set of questions can be placed in many different random orders, but the data will return in a fixed order for analysis. That trick was used by Birnbaum (2000b) in his FactorWiz program, which creates the HTML for studies with randomized orders of factorial combinations.

EXHIBIT 2.2

A simple Perl script that saves data (simple1.pl). This file is placed in the Web site folder called "cgi-bin." Data are saved into a file named "data.txt," which is inside a folder named "data." This script sorts data according to leading digits that precede the input variable names. Each datum appears in quotes, separated by commas. This type of file can be easily imported to spreadsheet programs such as Excel. The word *end* is placed at the end of each data record. The user is redirected to a thank-you message in the file thanks.htm.

```
#!C:/perl/bin/perl.exe
$path_to_datafile = ". ./www/data";
$redirect_to = ". ./thanks.htm";
use CGI;
$query = new CGI;
#open data file (in folder data, data.txt) and save data
open(INFO, ">>$path_to_datafile/data.txt");
foreach $key (sort($query->param))
{
$value = $query->param($key);
print INFO "\ "$value\",";
}
print INFO "\ "end\ "\n";
close (INFO);
print $query->redirect($redirect_to);
```

To learn how to install a generic Perl script to save data to your own server, read the instructions on the supplementary site for this chapter. This Perl script can be used with all of the examples in Birnbaum (2001).

Another CGI programming language is PHP (originally standing for Personal Home Page, now standing for PHP: Hypertext Preprocessor), a set of powerful hypertext processing tools used to create dynamic Web pages that can respond to and interact with the client. PHP is available from http://php.net/. A generic form processor in PHP is provided in Göritz and Birnbaum (2005), which also works with the examples in Birnbaum (2001).

Organization of a Web Site

Table 2.1 shows the organization of the bare bones Web site that has been created at the supplementary site for this chapter. One of the two main folders is called "www" and contains the HTML files that can be viewed via the Web. Depending on the server and its configuration, this folder might be called "htdocs" or "website," for example, instead of "www." It contains a home page called "index.htm." When a person enters the URL for this site in their browser, the file index.htm in this folder will be displayed.

TABLE 2.1

Organization of a Simple Web Site

www (folder)	cgi-bin (folder)
index.htm	simple.pl (see Exhibit 2.2)
Listing1.htm (Exhibit 2.1)	generic.pl
Survey_1.htm	
thanks.htm	
data (folder)	
data.txt (file inside the data folder)	

Note. This Web site has two main folders: www (which contains the HTML documents that can be viewed via the Web) and cgi-bin (which contains the CGI scripts that save data to the server). The file index.htm is the home page that will be displayed when a person types in the URL of the Web site. This page might contain text describing the study, inviting people to participate, and a link to the study. Two studies, survey1.htm and survey2.htm are listed. The file thanks.htm contains a "thank you" and debriefing message. It could also contain links to other pages (e.g., other studies to do, a report describing the research program). The data folder contains the data.txt file, to which the data are appended. There could be other data files as well. The data folder and its files can be password protected.

The www folder also contains two surveys, "Listing1.htm" and "survey_1.htm." These files are linked via hyperlinks from the home page, which would normally contain information about the studies and an invitation to participate. This folder also contains the file thanks.htm, to which the participant is directed when he or she completes the survey and submits the data. The thank you page might contain links to other surveys, to debriefing materials, or to other resources.

The other main folder is the cgi-bin folder, which in this case contains two scripts written in Perl. The first of these is "simple.pl," which contains Exhibit 2.2. Also included in this folder is the generic script, "generic.pl," that allows one to record the date and time of the submission and to gather the remote Internet Protocol (IP) address of the computer used by the participant, as well as other information. These scripts are saved as simple text files with the extension, ".pl" (i.e., ".PL"), which indicates a Perl program.

A researcher might recruit participants by various methods to visit the home page. From there, the person would be invited to click a link to the appropriate survey or experiment. When the person completes the study and clicks the "submit" button, the data are sent to the CGI (e.g., the generic Perl script), which saves the data in a file (in this case, it is called "data.txt" in a folder named "data"), and the participant is redirected to the page (in this case, "thanks.htm"), which would contain a "thank you" message and appropriate debriefing, including perhaps links to read more about the research topic. In the bare bones Web site, a link has been provided to the data file, so people testing the system can check that their data arrived correctly. (After scripts have been tested, the link to the data would normally be removed and the permissions on the data folder and data files would be set so that only the experimenter can view them by signing in with a password.)

What Server to Use?

The term *server* refers to both (a) the computer that houses and "serves" (delivers) the HTML files to the participant and (b) the computer software that performs those functions. It is possible that two servers might be used: One server might host the surveys and experiments, and the other might be used to save the data and redirect the participant to the "thank you" message or next part of the study. Usually, however, the same server would be used for both the survey and the programs that save data.

Perhaps your university or department maintains a server that can host your surveys and experiments. You may have people on your campus who can provide helpful and knowledgeable tech support. To upload your files to your campus server from a remote site (such as your

home computer, for example), you can use a File Transfer Protocol (FTP) program. This method requires that your department or university provide you an account and an FTP password. Many free programs perform FTP, which can be found by searching on sites such as http://www.download.com for *FTP* or using search engines such as http://www.google.com to find such sites for shareware and freeware. If you are allowed to work on the server directly, you can simply copy your files from a flash drive or other data storage device to your folder of Web files on the server.

To save data from Web surveys and experiments on a server, you create Web pages that collect and send data to a CGI script on the server that organizes and saves the data. To create surveys in HTML, you might use SurveyWiz (Birnbaum, 2000b), which is a very simple-to-learn program that creates the HTML for a survey consisting of text boxes, true–false, multiple choice, and rating scale items. To try out SurveyWiz, visit the supplementary Web site for this chapter. For your CGI script, you can use a Perl script such the one described above, "generic.pl." Alternatively, you might use the PHP generic form processor (Görtiz & Birnbaum, 2005).

Unfortunately, a problem that some researchers encounter at universities is that technicians who are placed in charge of servers may not be knowledgeable about Web research and may also be overly concerned about security or their own personal control of the site. You might not be allowed to work directly on the server, to see its log files, or to install Perl programs or other such tasks that might be needed for your research.

Sometimes technicians are asked to enforce directives from administrators who want to dictate the appearance or even the content of all Web pages hosted by the university. Some universities create rules about what can or cannot be put on "their" Web sites. Other universities require that half or more of the space of every Web page be filled with promotional banners and insignias of the university, to make everything look official and uniform.

To avoid such problems in dealing with universities that do not allow academic freedom on the Internet, one can use a private Web hosting service. Some of these providers offer basic Web hosting for free, which usually means that commercial banner ads will be placed in some, if not all, of the Web pages in a site. For example, in 2009, the company www.netfirms.com offered basic Web hosting free, including the option to include Perl scripts, as well as other features. To find this service, visit http://www.netfirms.com/web-hosting/web-hosting-basic/. My example site, at the supplementary Web site for this chapter, also includes instructions on installing a generic Perl script and illustrates how your site might look (including the advertisements) when hosted this way.

For a fee of about $5 per month, you could have a commercial Web site without the ads and conduct your research there. For many investigators, a commercial site costing $5 per month is well worth the expense.

Such an approach provides one freedom of expression such as is no longer available at many universities. Furthermore, you can relax: Others will make sure that the site is up and running and that it is secure, relieving you of the daily needs to make sure that the power has not gone out, that the system has not crashed, and that hackers are not attacking your site. No one will ask you to change fonts or to install university logos on all pages. You won't need permission from the technical staff every time you want to upload or change your files.

Some people, however, prefer to run their own server. This solution gives you maximal control over all of the files on your computer and is a preferred method if you will be using a database that interacts with your participants or requires a lot of server-side programs. This approach requires that you have a good connection to the Internet (through either your university or a commercial Internet service provider), you have server software, you know how to use these programs, and you can maintain the server and make sure that you keep everything running.

Fortunately, the best server software is free. The Apache HTTP Server is included (installed) with all new Macintosh computers. If you have a Mac, all you have to do is turn on the server that has already been installed and make a few simple adjustments, which are described in Birnbaum and Reips (2005). If you have a PC running Windows or Linux, you can download the free Apache software from http://www.apache.org/.

Göritz (2004) has described how to install a package of free software, including Apache HTTP Server, along with PHP and MySQL, which are used to collect, organize and store data in a database. See Göritz and Birnbaum (2005) for a generic script in PHP that can be used to emulate the scripts needed to work with Birnbaum's (2000a, 2001) examples. Links to Göritz's resources are given in the next section, which deals with the topic of programming the server.

Server-Side Programming

Server-side programs can feed dynamic material to the client on the basis of what the client puts in. The term "server-side" means that these programs run on the server rather than on the participant's computer. The two most popular server-side languages are PHP and Perl. The PHP Hypertext Preprocessor is a server-side programming language that allows Web developers to create dynamic content that interacts with databases. Like Perl, PHP is free but requires some study to learn to use its full powers. Neither you nor your participant need buy anything.

These are called "server-side" programming languages because the programs are executed by the server rather than by the client's com-

puter. That means that these programs will work for all participants, no matter what type of computer they have, as long as they can access the Internet and display Web pages.

Both of these free languages are described in tutorials that can be found on the Web. Books are also available on these languages. Perl is available from http://www.perl.org, and PHP can be found at http://www.php.net. For a tutorial on PHP, see http://us.php.net/tut.php.

Schwarz (1998) presented an introductory tutorial on Perl, and Fraley's (2004) book on Web experimentation contains lessons in Perl. Fraley showed how to use Perl to accomplish common tasks in behavioral research such as random assignment of participants to conditions and random ordering of items within a study. Many of his examples and other resources can be found at http://www.web-research-design.net/.

If you plan to collect surveys and experiments from participants and then simply analyze the results, you do not need any more than a generic script that saves the data to a file on the server (see, e.g., Exhibit 2.2). You do not need to run your own server. Your data can be saved in a simple data file that can be imported to statistical programs like SPSS.

Schmidt (1997) has written software called WWW Survey Assistant, which creates both a survey and a corresponding CGI script in Perl to collect data from surveys and save them to a server. For information on his approach, see http://www.mohsho.com/s_ware/how.html.

If you intend to interact with your research participants in a dynamic manner (or over an extended period of time), you will want to run your own server and use server-side programming to save data in a database. The most popular database software is MySQL (My Structured Query Language), which is also open source and free: http://www.MySQL.org. Open source means that program source code can be seen by everyone. An advantage of open source, free software is that thousands of people will test and evaluate the software and work together to improve it.

Database (as opposed to a simple *data file*) means an organized arrangement of information that can be added to, modified, or queried in various ways that can be automated. For example, suppose you have a long-term study in which people answer questionnaires and respond to questions over a period of several years. You want the database to keep track of participants, to automatically remind them if they have not completed what they should have completed by a certain date, and perhaps to compute scores on the basis of their answers and give feedback or to present selected follow-up questions contingent on a computation based on previous answers.

You might want to allow participants to complete part of a questionnaire and return later to complete the rest of it, and in this case you would want the computer to remind them where they left off. The database can hold information from previous surveys, and the server-side

software can inquire of the database whether the participant has completed everything he or she should have done by a given due date (which might depend on the date of the previous participation by the participant). To accomplish these goals, your best solution is to install the Apache server, PHP, and MySQL.

One can find installation packages and instructions on the Web. Göritz has used this method and has written tutorials on how to accomplish many of the useful tasks of longitudinal research. Her teaching materials can be found via the supplementary Web site for the book.

You might wish to keep track of a panel of participants (a subject pool or an online panel). Göritz has written some special programs in PHP that accomplish these tasks, which are free and available on the book's supplementary Web site.

Client-Side Programming

Besides HTML, which is delivered by the server but instructs the client's browser how to display the materials, there are three powerful approaches to programming the client's computer to run a survey or experiment: JavaScript, Java, and Authorware. Because these programs run on the client's side, the client must have something installed and turned on for them to work.

JAVASCRIPT

The first of these languages is JavaScript, which can be used to verify a participant's response, measure response times, randomize assignment to conditions, and do other tasks that require interaction with the participant. This language is supported by most browsers and is now used throughout the Web. In its early days, some users turned it off out of fear of the security lapse of allowing somebody else to run a program on their machines. Schwarz and Reips (2001) noted that when a task can be done equally well by the server or on the client's machine, it may be preferable to use the server-side program rather than lose people who do not have JavaScript installed. Of these three programming approaches, JavaScript is probably the one most likely to run on your participant's machine.

An introduction to basic JavaScript is given in Birnbaum (2001), who provided three chapters on this approach. There are many tutorials on the Web and books on the topic. Birnbaum and Wakcher (2002) showed how the language can be used to control an experiment on probability learning and present a basic tutorial with a series of examples, which are available at the supplementary Web site for this chapter.

JavaScript can be used in very small bits and pieces to perform simple tasks such as randomization of order of presentation of materials, random assignment of people to conditions, checking for reasonable responses or missing responses, measuring response time, and many other such tasks. For more information on JavaScript, visit the following URL: http://www.webreference.com/js/resources/.

JAVA

Java is a powerful object-oriented programming language. It should not be confused with JavaScript, even though these two languages have some similarities. Whereas JavaScript is typically included in a Web page in script form, programs written in Java are precompiled and saved as byte codes that are delivered to the client as *applets,* much like photographs or other media. The client must have the Java engine installed and turned on for the applet to work. The language allows good control of graphics, timing control, and time measurement. Java is a good approach for a person who is a good computer programmer or who is willing to put in the work to become one. Some illustrations of what can be done via Java are provided in McClelland's (2000) *Seeing Statistics:* http://psych.colorado.edu/~mcclella/java/zcalc.html and http://psych.colorado.edu/~mcclella/ATIJava/.

Francis, Neath, and Surprenant (2000) described how Java can be used to conduct many classic examples of cognitive psychology studies. These demonstrations as well as Java experiments on social psychology are now available from Wadsworth, at http://coglab.wadsworth.com/ and http://soclab.wadsworth.com/.

AUTHORWARE

Many of the same tasks that can be done through Java can also be done by means of the fairly expensive program, Authorware. Programming in Authorware involves pushing icons representing various actions such as timed visual displays, loops, computations, interactions with the user, and so on. A brief description of this approach is available at http://www.apa.org/science/psa/williams_prnt.html. Many studies in cognitive psychology created in Authorware are available from the Ole Miss site, created by McGraw, Tew, and Williams (2000), at the following URLs:

http://psychexps.olemiss.edu/
http://www.psych.uni.edu/psychexperiments/.

The use of Authorware software to run experiments requires that the participant has the Authorware player installed on his or her machine. That means that Authorware is an appropriate choice for experiments that can be done in a lab with computers on which the appropriate player

has been installed. Another drawback is that Adobe announced that it will support the existing software but cease development of Authorware.

Besides Java, JavaScript, and Authorware, there are other ways to use the client's computer to control the study. At the present time, I cannot recommend that people use software for this purpose, however. Besides the costs of such commercial software, Microsoft software is known for security problems, bugs, and incompatibilities with computer systems running rivals' software. When you conduct experiments via the WWW, you cannot force your participants to use Microsoft products (unless you run in the lab), so it is a strong limitation to use software that works properly only when your Web participants are using Microsoft products.

Web Research Tools

Reips, in association with several collaborators, has created a number of tools for the Web experimenter. These are available at http://psych-iscience.unizh.ch/index.html.

LogAnalyzer (Reips & Stieger, 2004) is a program for extracting information from server logs. When studies are set up as a factorial design, for example, with different treatment combinations in different Web pages, this program can organize and analyze data for analyses of variance, analyze drop-out rates and response times for each treatment condition, and explore many other aspects of the data.

WEXTOR (Reips & Neuhaus, 2002) helps experimenters create and organize the materials for complex experimental designs that can have within-subject and between-subjects conditions. This program also creates a graphical representation of the experimental or quasi-experimental design. Other resources for online experimentation are also available from the above site, including the opportunity to recruit participants from the Web Experimental Psychology Lab (Reips & Lengler, 2005).

Deciding What You Need

Table 2.2 presents a summary of the major techniques of Web-based research that can help you decide what you need to accomplish your type of research online. As noted earlier, anyone planning to do Web-based research needs to know basic HTML (including the technique of Web forms).

TABLE 2.2

Summary of Major Techniques of Web-Based Behavioral Research

Technique	Uses and considerations
HTML: A free programming language. An HTML file is a plain text file whose name has an extension of .htm or .html (e.g., MyWebPage.htm). You can construct the file in a plain text editor (e.g., Notepad) or in a Web page editor (e.g., Dreamweaver). Do not leave spaces in the names of these files; instead, you can use the underscore character (shift-minus), as in my_web_page.htm. If you plan to do research via the WWW, you need to learn basic HTML.	Basic to Web research. Presents and formats text, pictures, graphics, sounds, video, media. See Birnbaum (2001; chaps. 2, 3, 4). Many free Web sites contain good tutorials and summaries. Simple programming effects can be created by design of hyperlinks. For example, if you have different surveys for men and women, smokers and nonsmokers, and so on, you can use links to branch in a survey.
HTML forms: Part of HTML. The free programs SurveyWiz and FactorWiz create Web pages containing HTML forms that send the data to a generic CGI script, which organizes and saves the data sent via the form, and directs the participant to a thank-you page.	See Birnbaum (2001, chap. 5). See also the Perl script by Schmidt that emulates the generic script used by Birnbaum, called "generic.pl," which works with any of the HTML forms in Birnbaum's (2001) book.
SurveyWiz, FactorWiz: Free programs that allow you to make a simple questionnaire or factorial experiment quickly. They are Web pages that make Web pages that run experiments and surveys on the Web.	See Birnbaum (2000). These are easy to learn and easy to use. They are relatively restricted in what they do. You can mix questions requiring a short typed answer or numerical response, with questions with scales of radio buttons or multiple choice. They also allow blending with graphics to put almost any paper-and-pencil study on the Web.
WWW Survey Assistant: Free software, written by Schmidt (1997), that is more powerful than SurveyWiz, but it requires more effort to learn. Creates both the HTML and the Perl scripts to make computations on the data as well as save them to the server.	See links by Schmidt (1997), (http://www.mohsho.com/s_ware/how.html) which describe comparisons of this free software against commercial products.
WEXTOR: A free program that creates the HTML files needed for all sorts of experimental designs, especially between-subjects designs that use different Web pages for different parts of the study.	See Reips & Neuhaus (2002) and materials by Reips, Blumer and Neuhaus (http://psych-wextor.unizh.ch/wextor/en/index.php). Advantage of breaking up a study into smaller parts: You can study dropout in detail. WEXTOR creates a visual display of design and skeleton of the Web pages.
LogAnalyzer: Program that analyzes log files.	This program fits well with the WEXTOR approach, in that one can study all of the requests for files and places where dropouts occur.

(*continued*)

TABLE 2.2 (*Continued*)

Summary of Major Techniques of Web-Based Behavioral Research

Technique	Uses and considerations
Java: Free programming language that typically runs on client side. Neither you nor participant need buy anything. Participant must have it installed and turned on, which is true for most users.	Very powerful, object-oriented programming language. Can be used to make stand-alone programs or Web applets. Java has other uses as well, including precise control and measurement of graphics and events on the screen, such as the position of the mouse. A possible disadvantage is that Java is not installed client-side for some people.
JavaScript: Free programming language that runs on client side. Neither you nor the participant need buy anything, but participants must have it installed and turned on. The scripts can be included in the Web page, which makes your studies open source, which allows others to replicate exactly and build on your materials.	Powerful language. Can add little bits to Web pages to add functionality. Can make programs, including ones to control experiments, manipulate sequence, randomize, time, measure time, and so on. See Birnbaum (2001; chaps. 17–19). A possible disadvantage is that JavaScript is not installed client-side for some people.
Authorware: A commercial program that allows construction of dynamic experiments that run on the client's side. This approach is expensive has great power in creating experiments with control of timing, randomization, detection of screen events, insertion of media, and many other features. The participant must have installed the Authorware Player for the experiments to work. Best in the lab, or with an online panel of participants who have agreed to participate (e.g., they are paid) and have installed the player.	This technique can do many of the same things as Java but uses a graphic user interface with icons to control the experiment (McGraw, Tew, & Williams, 2000). The Ole Miss site (http://psychexps.olemiss.edu/) uses this method for its main power, but this approach also uses HTML and JavaScript as well as server-side programming, Excel macros, and other techniques. Study the manuals that come with Authorware program. A possible disadvantage is that Authorware player is not installed client-side for some people.
Server-side programming: Programming ran on your own Web server, including having CGI scripts installed to save data. Install script on your own server to save to a secure location. This works for all people.	See examples and materials by Göritz (2004). Server-side programming can also be used to do certain other tasks besides saving data (e.g., random assignment to conditions). This is the only method for anything requiring security, such as scoring an exam online or using passwords to sign in to the system.
Apache HTTP Server: Free software server. Göritz (2004) explained how to install this powerful program, which allows you to manage and run your own Web site(s). This gives you complete control over your experiments and data.	Running your own server might be an extra burden on your time and energy. For example, you may need to restart your machine after a system crash or power outage, which may be difficult if you travel a lot.

TABLE 2.2 (*Continued*)

Summary of Major Techniques of Web-Based Behavioral Research

Technique	Uses and considerations
Perl: A free programming language that can be used for server-side programming.	Perl can be used to write CGI scripts that save data from your survey or experiment on your server, for example.
PHP: A free method for server-side programming.	Göritz (2004) described this technique, which has become very popular. Perl and PHP can do many of the same tasks. See also Göritz and Birnbaum (2005).
MySQL: A package for database management, which can be dynamically linked to Web content. For example, one could compute statistics from a study and program it to automatically update online as more people participate. This package is also free.	Göritz (2004) discussed how to install and use MySQL. For some applications, you need only a simple script to save data in a file. However, for many dynamic tasks, the database can provide the solution. One can keep track of people who come and go to the site, remind them what they have and have not finished, and so on.

A person who plans to conduct surveys and Web-based experiments that are of the type that used to be done by pencil and paper can work effectively without needing to know much about how to operate or program a server. Using a server that is operated by a university department or a commercial service provider, one can conduct such research using a generic CGI script to save the data and redirect the participant to an appropriate page. The generic Perl script or the generic PHP forms processor can be used, along with tools such as SurveyWiz and FactorWiz or WWW Survey Assistant to create HTML pages that work with these resources.

Many tasks can be done either by client-side programs or server-side programs: random assignment to conditions, random ordering of the stimuli or items in a questionnaire, timing of displays, response time measurement, and interaction with the participant. Other tasks can be done only by server-side programming. For example, the saving of data to the server can be done only on the server side. Anything requiring the use of passwords, exam keys, and similar issues that require real security should be done only on the server side.

The best choices for free, client-side software are JavaScript and Java. Studies involving randomization of trials, probabilistic stimuli, probabilistic reinforcement, or control of temporal presentations can be done by either of these programming languages. For fine control of visual displays, Java is probably the better choice, although the programming language

can be more difficult to learn. Authorware is not free, but it allows programming by moving icons on a flowchart. Java programming is done through lines of text. Java applets are usually precompiled and sent to the participant's computer in the form of byte codes.

The most popular choices for free server-side software are Apache for the Web server, Perl or PHP for the CGI programming, and MySQL for the database software (Göritz, 2004; in press). These software products are open source and free. These techniques are needed when you plan to manage panels of participants, allow participants to interact with each other, keep track of performance by participants over a long period of time, and perform other such dynamic and interactive tasks.

References

Birnbaum, M. H. (Ed.). (2000a). *Psychological experiments on the Internet.* San Diego, CA: Academic Press.

Birnbaum, M. H. (2000b). SurveyWiz and FactorWiz: JavaScript Web pages that make HTML forms for research on the Internet. *Behavior Research Methods, Instruments, & Computers, 32,* 339–346.

Birnbaum, M. H. (2001). *Introduction to behavioral research on the Internet.* Upper Saddle River, NJ: Prentice Hall.

Birnbaum, M. H., & Reips, U.-D. (2005). Behavioral research and data collection via the Internet. In R. W. Proctor & K.-P. L. Vu (Eds.), *Handbook of human factors in Web design* (pp. 471–491). Mahwah, NJ: Erlbaum.

Birnbaum, M. H., & Wakcher, S. V. (2002). Web-based experiments controlled by JavaScript: An example from probability learning. *Behavior Research Methods, Instruments, & Computers, 34,* 189–199.

Fraley, R. C. (2004). *How to conduct behavioral research over the Internet: A beginner's guide to HTML and CGI/Perl.* New York: Guilford Press.

Francis, G., Neath, I., & Surprenant, A. M. (2000). The cognitive psychology online laboratory. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 267–283). San Diego, CA: Academic Press.

Göritz, A. S. (2004). Apache, MySQL, and PHP for Web surveys. http://www.goeritz.net/ati/ [WWW Document.]

Göritz, A. S. (in press). Building and managing an online panel with phpPanelAdmin. *Behavior Research Methods.*

Göritz, A. S., & Birnbaum, M. H. (2005). Generic HTML form processor: A versatile PHP script to save Web-collected data into a MySQL database. *Behavior Research Methods, 37,* 703–710.

Krantz, J. H. (2001). Stimulus delivery on the Web: What can be presented when calibration isn't possible? In U.-D. Reips & M. Bosnjak

(Eds.), *Dimensions of Internet science* (pp. 113–130). Lengerich, Germany: Pabst Science.

Krantz, J. H., & Dalal, R. (2000). Validity of Web-based psychological research. In M. H. Birnbaum (Eds.), *Psychological experiments on the Internet* (pp. 35–60). San Diego, CA: Academic Press.

McClelland, G. (2000). *Seeing statistics.* Duxbury Press.

McGraw, K. O., Tew, M. D., & Williams, J. E. (2000). PsychExps: An online psychology laboratory. In M. H. Birnbaum (Eds.), *Psychological experiments on the Internet.* (pp. 219–233). San Diego, CA: Academic Press.

Reips, U.-D., Blumer, T., & Neuhaus, C. (2009). WeXtor 2.5: Develop, manage, and visualize experimental designs and procedures. [WWW document]. http://psych-wextor.unizh.ch/wextor/en/index.php

Reips, U.-D., & Lengler, R. (2005). The *Web experiment list:* A Web service for the recruitment of participants and archiving of Internet-based experiments. *Behavior Research Methods, 37,* 287–292.

Reips, U.-D., & Neuhaus, C. (2002). WEXTOR: A Web-based tool for generating and visualizing experimental designs and procedures. *Behavior Research Methods, Instruments, & Computers, 34,* 234–240.

Reips, U.-D., & Stieger, S. (2004). Scientific LogAnalyzer: A Web-based tool for analyses of server log files in psychological research. *Behavior Research Methods, Instruments, & Computers, 36,* 304–311.

Schmidt, W. C. (1997). World-Wide Web survey research made easy with WWW Survey Assistant. *Behavior Research Methods, Instruments, & Computers, 29,* 303–304.

Schwartz, A. (1998). Tutorial: PERL, a psychologically efficient reformatting language. *Behavior Research Methods, Instruments, & Computers, 30,* 605–609.

Schwarz, S., & Reips, U.-D. (2001). CGI versus JavaScript: A Web experiment on the reversed hindsight bias. In U.-D. Reips & M. Bosnjak (Eds.), *Dimensions of Internet science* (pp. 75–90). Lengerich, Germany: Pabst Science Publishers.