27 Behavioral Research and Data Collection via the Internet

Ulf-Dietrich Reips and Michael H. Birnbaum

CONTENTS

In the past 15 years it has become possible to collect data from participants who are tested via the Internet rather than in the laboratory. Although this mode of research has some inherent limitations owing to lack of control and observation of conditions, it also has a number of advantages over lab research. Many of the potential advantages have been well-described in a number of publications (Birnbaum 2000a, 2001a, 2004a, 2007; Krantz and Dalal 2000; Reips 1995, 2000, 2006, 2007; Reips and Bosnjak 2001; Schmidt 1997a, 1997b). Some of the chief advantages are that (1) one can test large numbers of participants very quickly, (2) one can recruit large heterogeneous samples and people with rare characteristics, and (3) the method is more cost-effective in time, space, and labor in comparison with lab research. This chapter provides an introduction to the major features of the new approach and illustrates the most important techniques in this area of research.

27.1 OVERVIEW OF INTERNET-BASED RESEARCH

The process of Web-based research, which is the most frequent type of Internet-based research, can be described as follows: Web pages containing surveys and experiments are placed in Web sites available to participants via the Internet. These Web pages are hosted (stored) on any server connected to the World Wide Web. People are recruited by special techniques to visit the site. People anywhere in the world access the study and submit their data, which are processed and stored in a file on a secure server. (The server that "hosts" or delivers the study to the participant and the server that receives, codes, and saves the data are often the same computer, but they can be different.)

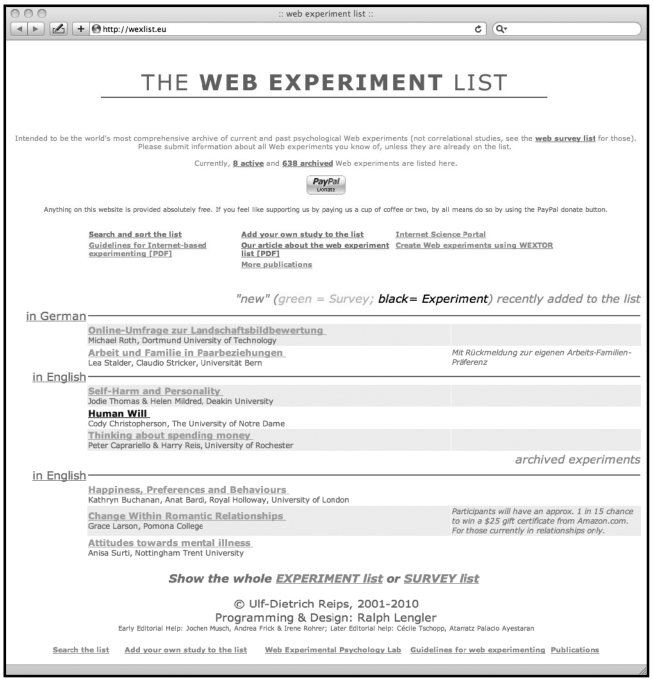
The Internet scientist plans the study following guidelines while striving to avoid pitfalls (Birnbaum 2001a, 2004a, 2004b, 2007; Reips 2002b, 2002c, 2007; Reips and Bosnjak 2001; Schmidt 2007). The researcher creates Web pages and other files containing text, pictures, graphics, sounds, or other media for the study. He or she will upload these files to the host server (as needed) and configure the Web server to accept, code, and save the data. The researcher tests the system for delivering the experiment and for collecting, coding, and saving the data. The Web researcher must ensure that the process is working properly, recruit participants for the study, and finally retrieve and analyze the data. Although this process may sound difficult, once a researcher has mastered the prerequisite skills, it can be far more efficient than traditional lab methods (Birnbaum 2001a; Reips 1995, 1997, 2000; Reips and Krantz 2010).

27.2 PSYCHOLOGICAL RESEARCH ON THE WEB

To get an overall impression of the kinds of psychological studies that are currently in progress on the Web, visit studies linked at the following sites:

Web experiment list (Reips and Lengler 2005): http://wexlist.net (see Figure 27.1)
Psychological Research on the Net: http://psych.hanover.edu/research/exponnet.html
Web Experimental Psychology Lab (Reips 2001a, 2001b): http://wexlab.eu
Decision Research Center: http://psych.fullerton.edu/mbirnbaum/decisions/thanks.htm

The number of studies conducted via the Web appears to have grown exponentially since 1995 (Reips 2007), when psychologists began to take advantage of the new standard for HTML that allowed for convenient data collection (Musch and Reips 2000). Internet-based research has become a new topic and a new set of methods in psychology. The basics of authoring Web-based research studies will be described in the next sections.



THE **WEB EXPERIMENT** LIST

Intended to be the world's most comprehensive archive of current and past psychological Web experiments (not correlational studies, see the **web survey list** for those). Please submit information about all Web experiments you know of, unless they are already on the list.

Currently, **8 active** and **638 archived** Web experiments are listed here.

Anything on this website is provided absolutely free. If you feel like supporting us by paying us a cup of coffee or two, by all means do so by using the PayPal donate button.

Search and sort the list
Guidelines for Internet-based experimenting [PDF]
Add your own study to the list
Our article about the web experiment list [PDF]
More publications
Internet Science Portal
Create Web experiments using WEXTOR

"new" (green = Survey; black= Experiment) recently added to the list

in German

Online-Umfrage zur Landschaftsbildbewertung
Michael Roth, Dortmund University of Technology

Arbeit und Familie in Paarbeziehungen
Lea Stalder, Claudio Stricker, Universität Bern
Mit Rückmeldung zur eigenen Arbeits-Familien-Präferenz

in English

Self-Harm and Personality
Jodie Thomas & Helen Mildred, Deakin University

Human Will
Cody Christopherson, The University of Notre Dame

Thinking about spending money
Peter Caprariello & Harry Reis, University of Rochester

archived experiments

in English

Happiness, Preferences and Behaviours
Kathryn Buchanan, Anat Bardi, Royal Holloway, University of London

Change Within Romantic Relationships
Grace Larson, Pomona College
Participants will have an approx. 1 in 15 chance to win a $25 gift certificate from Amazon.com. For those currently in relationships only.

Attitudes towards mental illness
Anisa Surti, Nottingham Trent University

Show the whole EXPERIMENT list or SURVEY list

© Ulf-Dietrich Reips, 2001-2010
Programming & Design: Ralph Lengler
Early Editorial Help: Jochen Musch, Andrea Frick & Irene Rohrer; Later Editorial help: Cécile Tschopp, Atarratz Palacio Ayestaran

Search the list **Add your own study to the list** **Web Experimental Psychology Lab** **Guidelines for web experimenting** **Publications**

FIGURE 27.1 (See color insert following page xxx.) The Web experiment and Web survey lists at http://wexlist.net.

27.3 CONSTRUCTING STUDIES FOR THE INTERNET

There are many computer programs that allow one to create Web pages without knowing HTML. These programs include Adobe GoLive, Adobe Contribute, Adobe Dreamweaver, and Microsoft FrontPage (not recommended), among others. In addition, programs intended for other purposes, such as Open Office, Microsoft Word, PowerPoint, and Excel, or Apple Pages and Keynote allow one to save documents as Web pages. Although these programs can be useful on occasion, those doing Web research really need to understand and be able to compose basic HTML. While learning HTML, it is best to avoid these authoring programs. If you already know how to use these programs, you can study HTML by using them in *source code mode*, which displays the HTML rather than the "what you see is what you get" (WYSIWYG) or "layout" display.

There are many free, useful tutorials on the Web for learning about HTML and many good books on the subject. Birnbaum (2001a, chaps. 2–4) covers the most important tags (basic units in HTML) in three chapters that can be mastered in a week, with a separate chapter (Chapter 5) for the technique of Web forms, which is the technique that made Web research practical when this technique was supported by HTML 2, introduced in late 1994.

27.3.1 Web Forms

There are three aspects of Web forms that facilitate Internet-based research. First, forms support a number of devices by which the reader of a Web page can send data back to the server chosen by the author of a page. Forms support two-way communication of information, with the possibility for dynamic communication.

Second, Web forms allow a person without an e-mail account to send information from a computer, even if the computer is not configured to send e-mail. For example, a person at a local library, in an Internet café, or in a university lab could fill out a Web form on any Internet-connected computer, and click a button to send the data. This means that participants can remain anonymous.

Third, Web forms can deliver their data to a program on the server that codes and organizes the data and that saves them in a convenient form for analysis. In fact, server-side programs can even analyze data as they come in and update a report of cumulative results.

The Web form is the HTML between and including the tags, <FORM> and </FORM>, within a Web page. The response or "input" devices supported by forms allow the users (e.g., research participants) to type in text or numerical responses, click choices, choose from lists of selections, and send their data to the researcher. Table 27.1 shows a very simple Web form. You can type this text, save it with an extension of ".htm," and load it into a browser to examine how it performs. Table 27.1, along with other examples and links, are available from the following Web site, which is associated with this chapter: http://wexlab.eu/hcihandb/ (with a mirror at http://psych.fullerton.edu/mbirnbaum/handbook/).

In this example, there are four input devices: a "hidden" value, an input text box, a "submit" button, and a "reset" button. The "hidden" input records a value that may be used to identify the data; in this case, the value is "MyTest1." The "value" of the "submit" button or "reset" button is what is displayed on the buttons, but the "value" of a text box is whatever the viewer types in that field. When the "reset" button is clicked, the form is reset; i.e., any responses that were typed in or clicked are erased.

When the "submit" button is clicked, the *action* of the form is executed. In this example, the *action* sends e-mail with the two variables to the e-mail address specified. You should change this to your own e-mail address, load the form in the browser, fill in your age, and click the submit button. If your computer and browser are configured to send e-mail, you will receive an e-mail message with your responses in the message. The encryption type attribute can be erased (save the file and reload it in the browser), and you will see the effect that this attribute has on how the e-mail appears.

TABLE 27.1
Bare Bones Web Form (Example 2.1)

```
<html>
<head>
<title>
My First Form
</title>
</head>
<body>
<form method=”post” action=”mailto:mbirnbaum@fullerton.edu”
  enctype=”text/plain”>
<input type=”hidden” name=”00exp” value=”MyTest1”>
1. What is your age?
<input type=”text” name=”01age” size=”4” maxlength=”6”>
<input type=”submit” value=”Send the Data”>
<input type=”reset” value=”start over”>
</form>
</body>
</html>
```

27.3.2 Server-Side Scripting to Save the Data

Although sending data by e-mail may be useful for testing Web forms or for small efforts, such as collecting RSVPs for a party, it is neither practical nor secure to collect large amounts of data via e-mail. Instead, we can let the Web server write the data to its log file for later analysis (Section 27.7), or we can use a computer program to code the data and save them in a file, in a form ready for analysis (Section 27.6). To do this, we use a Common Gateway Interface (CGI) script, e.g., written in Perl or PHP, that codes, organizes, and saves the data safely to a secure server (see Schmidt 1997a, 2000, 2007). The ACTION of the form is then changed to specify the URL address of this script.

For example, revise the FORM tag in Table 27.1 as follows:

```
<FORM METHOD='post' ACTION='http://
psych.fullerton.edu/cgi-win/polyform.
exe/generic'>
```

In this example, the ACTION specifies an address of a script that saves the data to a file named data.csv on the psych.fullerton.edu server. The script residing at this address is a generic one that accepts data from any form on the Web, and it arranges the data in order of the two leading digits in each input variable's NAME. It then redirects the participant to a file with a generic "thank you" message. This example is for instruction, demonstration, and testing purposes only. For real data collection, one should put the data file in a protected folder where it is only available to the researchers.

27.3.3 Downloading Data by FTP

To view the data, one can use the following link in a browser that supports File Transfer Protocol (FTP) or the more secure sFTP. This link specifies an FTP site with a username of "guest" and password of "guest99":

ftp://guest:guest99@psych.fullerton.edu

From this FTP site, you can download the file named data .csv. This file can be opened in a text editor or in Excel, among other applications. At or near the end of the file will appear a line that contains the "hidden" value ("MyTest1") and the datum that you typed in for age.

27.3.4 Obtaining and Using a Dedicated FTP Program

Although most browsers support FTP, it is more convenient to use a program dedicated to FTP that supports additional features. There are several FTP and sFTP (secure FTP) programs that are free to educational users, such as Fetch for the Mac and WS FTP LE for Windows PCs. These free programs can be obtained from CNET Download.com, which has the following URL: http://www.download.com

File transfer programs, FTP or sFTP, are not only useful for downloading data from a server, but they can also be used to upload files to a server, in the case of a server administrated by another person or institution. In a later section, we describe advantages of installing and running your own server. However, many academic users are dependent on use of a department or university server. Others have their Web sites hosted by commercial Internet service providers (ISP). In these cases, the academic researcher will upload his or her Web pages by means of FTP or a server-side program (e.g., a control panel) to the server and download data via sFTP, or WebDAV from the server. Like sFTP, WebDAV (Web-based Distributed Authoring and Versioning) allows for a secure transmission, so for transfers of the data one should always use one of these options.

Some academics find that they can get greater independence and control of their research by using a commercial ISP, rather than using a shared university server. At some campuses, the administrators or techs who control the server are very reluctant to allow faculty to collect data or to view the log files. Commercial sites can provide either free hosting (which may entail inclusion of commercial advertisements in the Web pages) or low cost service (typically about $5/month) without the advertisements. An example with educational materials of how to construct a Web site at a commercial ISP is provided in the following site:

http://ati-birnbaum.netfirms.com/

For this site, the URL of a generic script is as follows:

http://ati-birnbaum.netfirms.com/cgi-bin/generic.pl

This can be used in place of the ACTION in Section 27.3.2. The data will be added to the following site:

http://ati-birnbaum.netfirms.com/data/data.txt

Because the data in this case are freely available, this example is for instructional, demonstration, and testing purposes only. In real research, the file containing data should be protected, especially when data are not anonymous.

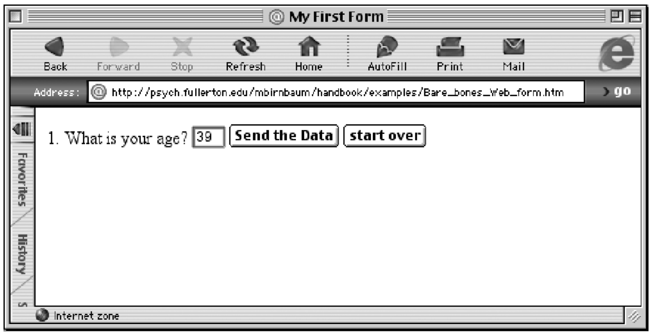
27.3.5 The "Hidden" Input Device

The display in the browser (Figure 27.2) shows the text in the body of the Web page, the text input box, *submit* button, and *reset* button. Note that the "hidden" input device does not display anything; however, one can view it by selecting Source (View Source, Frame Source or Page Source) from the View menu of the browser, so it would be a mistake to think that such a hidden value is truly hidden.

The term hidden unfortunately has the connotation that something "sneaky" is going on. The second author was once asked about the ethics of using hidden values in questionnaires, as if we were secretly reading the participant's subconscious mind without her knowledge or consent. In reality, nothing clandestine is going on. Hidden variables are routinely used to carry information such as the name of the experimental condition from one page to the next, to hold information from a JavaScript program that executes an experiment, or to collect background conditions such as date and time that the experiment was completed. In the current example, the hidden variable is used simply to identify which of many different questionnaires is associated with this line of data. This value can be used in Excel, for example, to segregate a mixed data file into subfiles for each separate research project (Birnbaum 2001a).

27.3.6 Input Devices

In addition to the text box, which allows a participant to type in a number or short answer, there are four other popular input devices, and one that makes use of images. The *textarea* input device is a rectangular box suitable for obtaining a longer response such as a paragraph or short essay. For multiple-choice answers, there are radio buttons, pull-down selection lists, and check boxes. Check boxes (usually displayed as squares on Web pages), however, should not be used in behavioral research. The problem with a check box is that it has only two states—it is either checked or unchecked. If a check box is unchecked, one does not know if the participant intended to leave it unchecked or perhaps just skipped over the item. For a "yes" or "no" answer, one must allow



My First Form

Address: http://psych.fullerton.edu/mbirnbaum/handbook/examples/Bare_bones_Web_form.htm

1. What is your age? 39

FIGURE 27.2 Appearance of basic Web form in the browser window.

at least three possibilities: “yes,” “no,” and “no answer.” In some areas of survey research, one may need to distinguish as many as five distinct response options for a yes/no question, “yes,” “no,” “no response,” “don’t know,” and “refuse to answer.” Reips et al. (2010) studied the impact of the presence of such nonresponse options on active nonresponse like “I don’t want to answer” and passive nonresponse (skipping). Combined active and passive nonresponse varied between 0 and 35% for the same item, when sensitive questions were asked.

We recommend avoiding the use of check boxes, except in situations where multiple responses to options in a list can be expected from a respondent, e.g., “check all music genres you were listening to today.” Another use is when a person is required to check a box in order to continue with the study, as in “Check here to indicate that you have read the information and agree to participate in this study.” Multiple-choice questions (including true-false questions) can be better handled by radio buttons than by check boxes.

With radio buttons (usually displayed as circles with round “dots” when clicked), one can construct a multiple-choice response device that allows one and only one answer from a potential list. The basic tags to create a yes/no question with three connected buttons are as follows:

```
<INPUT TYPE=radio NAME=02v2 VALUE=″″
checked>Do you drive a car?
<BLOCKQUOTE>
<BR><INPUT TYPE=radio NAME=02v2
VALUE=″0″>No.
<BR><INPUT TYPE=radio NAME=02v2
VALUE=″1″>Yes.
</BLOCKQUOTE>
```

In this example, the first radio button will be already checked, before the participant responds. If the participant does not respond, the value sent to the data is empty (there is nothing between the quotes); SPSS, Excel, and certain other programs treat this null value as a missing value. To *connect* a set of buttons, as in this example, they must all have the same *name* (in this example, the name is 02v2). When one button is clicked, the dot jumps from the previously checked button (the nonresponse, or null value) to the one clicked. If the respondent clicks “No,” the data value is “0” and if the respondent clicks “yes,” the data value is “1.”

We suggest that you follow this convention for yes/no responses: use larger numbers for positive responses to an item. In this case, the item measures driving. This convention helps prevent the experimenter from misinterpreting the signs of correlation coefficients between variables.

The selection list is another way to present a multiple choice to the participant, but this device is less familiar to both researchers and participants. Selection lists are typically arranged to initially display only one or two options, which when clicked, will expand to show other alternatives. The list remains hidden or partially revealed until the participant clicks on it, and the actual display may vary depending on how far the participant has scrolled before clicking the item. In addition, there are really two responses that the participant makes in order to respond to an item. The participant must drag a certain distance and then release at a certain choice. Alternatively, depending on the Web browser and how the Web page was created, the participant may type a letter and jump to the first item beginning with that letter. Because of the complexities of the device, precautions with selection lists are recommended.

First, like any other multiple-choice device, it is important not to have a legitimate response preselected but rather to include an option that says something like, “choose from this list” and that returns a “missing” code unless the participant makes a choice (Birnbaum 2001a; Reips 2002b). If a legitimate answer has been preselected, as in the left side of Figure 27.3, the experimenter will be unable to distinguish real data from those that result when participants fail to respond to the item. The illustration on the right side of the figure shows a better way to handle this list. Reips (2002b) refers to this all too common error (of preselected, legitimate values) as “configuration error V.”

Another problem can occur if the value used for missing data is the same as a code used for real data. For example, the second author found a survey on the Web in which the participants were asked to identify their nationalities. He noted that the same code value (99) was assigned to India as to the preselected “missing” value. Fortunately, the investigator was warned and fixed this problem before much data had been collected. Otherwise, the researcher might have concluded that there had been a surprisingly large number of participants from India.

Second, part of the psychology of the selection list is how the choice set and arrangement of options is displayed to the participant. It seems plausible that options that require a long scroll from the preset choice would be less likely to be selected. The experimenter communicates to the participant by the arrangement of the list (e.g., Smyth et al. 2006) and by the availability and placement of nonresponse options relative to the others (Reips et al. 2010). Birnbaum (2001a, Chap. 5) reported an experiment showing that mean responses can be significantly affected by the choice of options listed in a selection list. Birnbaum compared data obtained for the

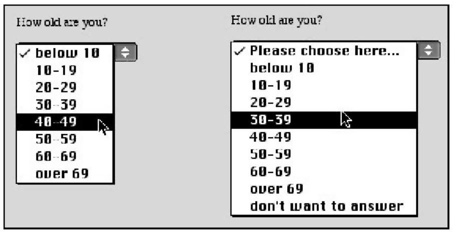


FIGURE 27.3 (left) Improper drop-down selection list and (right) a better way. The preselected option should be neutral, and the respondent should be given an option to express a deliberate nonresponse. Source: Reips (2002b).

judged value of the St. Petersburg gamble from three groups of participants who received different selection lists or a text input box for their responses. The St. Petersburg gamble is a gamble that pays $2 if a coin comes up heads on the first toss, $4 if the first toss is tails and the second is heads, $8 for tails-tails-heads, and so on, doubling the prize for each additional time that tails appears before heads, ad infinitum. One group judged the value of this gamble by means of a selection list with values spaced in equal intervals, and the other had a geometric series, with values spaced by equal ratios. A third group had a text box instead of a selection list and requested the participant to respond by typing a value. The mean judged value of this gamble was significantly larger with the geometric series than with equal spacing; furthermore, these values differed from the mean obtained with the text box method. The results therefore depend on the context provided by the response device.

In laboratory research on context effects (Parducci 1995), it has been shown that the response that one assigns to a stimulus depends on two contexts: the context of the stimuli (the frequency distribution and spacing of the stimulus levels presented) and the context provided by the instructions and response mode. Parducci (1995) summarized hundreds of studies that show that the stimulus that is judged as "average" depends on the endpoints of the stimuli, their spacing, and their relative frequencies. Hardin and Birnbaum (1990) showed that the response one uses to evaluate a situation depends on the distribution of potential responses incidentally shown as examples. It might seem therefore that one should try to "avoid" contextual effects by providing no other stimuli or responses; however, the head-in-the-sand approach yields even more bizarre findings.

To show what can happen when an experimenter tries to "avoid" contextual effects, Birnbaum (1999a) randomly assigned participants to two conditions in a Web study. In one condition, participants judged "how big" is the number 9, and in the other condition, they judged how big is the number 221. He found that 9 is significantly "bigger" than 221. Birnbaum (1999a) predicted that the result would occur based on the idea that each stimulus carries its own context, and even though the experiment specified no context, the participants supplied their own. So, one cannot avoid contextual effects by experimental design only.

In a study with (short) selection menus, Reips (2002a) found no significant difference between "drop-down" (preselected choice at top) versus "pop-up" (preselected choice at bottom) menus in choices made on the menu. Nevertheless, it would be dangerous to assume that this finding (of nonsignificance) guarantees immunity of this method from this potential bias, particularly for long selection lists.

Because the list of options, their spacing, order, and relative position of the initial value may all affect the results, we recommend that the selection list be used only for obtaining responses when the options are nominal, when there is a fixed list of possible answers, and the participant knows the answer by heart. For example, one might use such a device to ask people in what nations they were born. Although the list is a long one, one hopes that people know the right answer and will type the first letters correctly or patiently scroll to the right spot. Some investigators list their own country at the top of the list as well as at its place in the alphabetical sequence in order to facilitate this frequently chosen response in studies with participants from their own country.

There are many potential human factors that could be studied in selection lists, and much about their potential contextual effects is still unknown (see Birnbaum 2001b; Dillman 2001; Reips 2002a; Smyth et al. 2006, for early investigations, and Dillman, Smyth, and Christian 2008; Reips 2010, for more recent summaries).

One remaining category of input device to be discussed is that in which participants click on an image. This device allows the capture of the *x* and *y* coordinates of the spot on the image where the click occurred. One frequently used example is the visual analog scale. Visual analogue scales use a horizontal or vertical line segment. Participants are asked to click at a point that represents their judgment on a continuum defined by terms labeling the endpoints. For example, a participant may be asked to mark her degree of agreement with a political statement on a continuum ranging from "total disagreement" to "total agreement."

For the purpose of testing Internet-based visual analogue scales, Reips and Funke (2008) asked participants to repeatedly mark the points where 13 probabilities should be located on a 0% to 100% probability line segment (see also Funke and Reips 2007). They found that the average responses were approximately correct and linearly related to the nominal percentages, even for extreme scale lengths of 50 and 800 pixels. In previous research with this response mode using the subtractive model, it has been concluded that the mean responses are a sigmoidal function of subjective value (e.g., Rose and Birnbaum 1975; Stevenson 1993).

In early research using paper and pencil methods, the position on the line had to be measured individually for each response. This measurement can be automated by computer (e.g., Stevenson 1993). A description of the simple image method is given by Fraley (2004), and a method to construct such response scales using JavaScript is given by Baron and Siepmann (2000). They can even be constructed without the use of JavaScript, in pure HTML. See below an example by Reips (1999) from Internet-based research with children and adults, in which participants were asked to click on a "tunnel" (the visual analogue scale) at the point where "a snake's head would now be that had crawled into the tunnel." The scale was constructed by repeatedly linking the same image "pixel.gif" to the next page "next.html," each time with a different value "point" (here "37" and "38"). For a visual analogue scale with 100 pixels the code below would be repeated for values from 1 to 100 (full experiment available on the handbook Web site):

```
<A href="next.html?point=37"><IMG
src="pixel.gif" border="0"></A>
<A href="next.html?point=38"><IMG
src="pixel.gif" border="0"></A>
```

Reips and Funke (2008) developed VAS Generator (http://vasgenerator.net), a free tool to create visual analogue scales for Internet-based research. This tool automates construction of the response scales without requiring the user to do programming, and it allows the participant to mark and adjust the response before recording it.

Some findings on the "quality" of data obtained by browser-based methods are available (see e.g., Tuten, Urban, and Bosnjak 2002). Birnbaum (1999b) presented the same browser-based questionnaire on two occasions to 124 undergraduates in the laboratory. In the main body of the experiment, participants were to choose between pairs of gambles by clicking a radio button beside the gamble in each pair that they would prefer to play. In this part of the experiment, people agreed on average in 82% of their choices. With a multiple-choice response for gender, 2 of 124 participants switched gender from the first to second occasion; one switched from male to female and one made the opposite switch. In the choices between gambles, we assume that people were unsure of their choices or changed their minds, but in case of gender, it seems clear that people made errors. Every person agreed on his or her age, which was typed into a text box on each occasion. However, six people gave different answers for the number of years of education, which was also typed into a box.

Once it is realized that people can make errors with any response device, we see the need to design Web studies to allow for such inconsistent behavior. For example, suppose a researcher wants to study smoking behavior and suppose there are different questionnaires for people who are smokers, for those who never smoked, and for those who quit smoking and no longer smoke. If a participant makes an errant click on a branching question, that person might be sent to the wrong questionnaire, and most of the subsequent questions may be inappropriate for that person.

One approach to this problem of "human errors" is to build some redundancy and cross-examination into questionnaires and methods of linking people to different instruments. One device is the JavaScript prompt that provides cross-examination when a person clicks a link to identify his or her gender. If the person clicks "male," the prompt opens a new box with the question, "are you sure you are male?" requiring a yes/no answer before the person can continue. A similar check cross-examines those who click "female." A second technique is to provide paths that provide a "second chance" to link to the correct gender. Here, the person who clicks "male" then receives a page with a link to click "if you are female," which would send the person to the female questionnaire, which also has a second chance to revert.

Another approach has just been presented by Stieger and Reips (2010), who recorded via an application named "User Action Tracer" what Internet participants were actually doing on the study's Web page. They collected and recorded actions with their exact position (x and y coordinates): clicks (for all mouse buttons); double-clicks; clicks on checkboxes, radio buttons, list boxes, and submit buttons; choices in drop-down selection menus; inserted text in text boxes; keys pressed on the keyboard; and the position of the mouse pointer every half a second. This type of data is called "paradata" and it can be used to verify user entries.

27.4 CREATING SURVEYS AND EXPERIMENTS WITH SIMPLE DESIGNS

Typing HTML for a questionnaire can be tedious, and typing errors in HTML can introduce errors that would make it impossible to get meaningful data. Therefore, any Web page designed to collect data should be thoroughly tested before it is placed on the Internet for data collection. Birnbaum's (2000b) SurveyWiz and FactorWiz are programs that were written to help researchers avoid making mistakes in HTML coding. These programs are freely available on the Web, and they create basic sets of radio buttons and text boxes that will properly code and return the data. Still, when editing HTML files (for example, by importing them into Microsoft Word or by copying and pasting items without properly changing the names), there is potential for introducing errors that can ruin a study. More about checks to be conducted before collecting data on the Web will be presented in Section 27.8.

The leading digits on the name attribute are used by the default, generic CGI script that organizes and saves the data for SurveyWiz and FactorWiz. This particular script requires that leading digits on variable names be sequential and start at 00. However, the HTML need not be in that order. That means that one could cut and paste the HTML to rearrange items, and the data will still return to the data file in order of the leading digits (and not the order within the HTML file of the items). This device is also used by Birnbaum's (2000b) FactorWiz program. FactorWiz creates random orders for presentation of within-subjects factorial designs. Although the items can be put in as many random orders as desired, the data always return in the same, proper factorial order, ready for analysis by ANOVA programs.

27.4.1 USING SURVEYWIZ

Instructions for using SurveyWiz and FactorWiz are given by Birnbaum (2000b, 2001a) and within their files on the Web, which can be accessed from the following:

http://psych.fullerton.edu/mbirnbaum/programs/

SurveyWiz3 provides a good way to learn about making a survey for the Web. It automatically prepares the HTML for text answers and rows of radio buttons. One can add a preset list of demographic items by clicking a single button. The program is easy to use and is less likely to lead to errors than complex commercial programs. For example, suppose we want to calculate the correlation between the number of traffic accidents a person has had and the person's rated fear while driving.

In SurveyWiz, one simply enters the survey name and short name and then types the questions, one at a time. In

TABLE 27.2
Web Form Illustrating Input Devices

```
<HTML>
<HEAD>
<TITLE>Driving Survey</TITLE>
</HEAD><BODY BGCOLOR=’papayawhip’>
<FONT FACE=Arial>
<H3> Instructions for Driving Survey</H3>
<BR>Please answer these questions honestly.
<BR>
<HR>
<FORM METHOD=’post’ ACTION=’http://psych.fullerton.edu/cgi-win/
  polyform.exe/generic’>
<INPUT TYPE=hidden NAME=00exp VALUE=driving_srvy1 >
<INPUT TYPE=hidden NAME=01Date VALUE=pfDate>
<INPUT TYPE=hidden NAME=02Time VALUE=pfTime>
<INPUT TYPE=hidden NAME=03Adr VALUE=pfRemoteAddress>
<P>1. In how many accidents have you been involved when you were the
  driver?
<INPUT type=text NAME=04v1 SIZE=8 MAXLENGTH=20><BR>
<P><INPUT TYPE=radio NAME=05v2 VALUE=’’ checked>
2. I often feel fear when driving.<BR>
strongly disagree
<INPUT TYPE=radio NAME=05v2 VALUE=1>
<INPUT TYPE=radio NAME=05v2 VALUE=2>
<INPUT TYPE=radio NAME=05v2 VALUE=3>
<INPUT TYPE=radio NAME=05v2 VALUE=4>
<INPUT TYPE=radio NAME=05v2 VALUE=5>
strongly agree<BR>
<HR>
<INPUT TYPE=radio NAME=06sex value=’0’ CHECKED>
3. Are you Male or Female?<BR>
<BLOCKQUOTE>
<INPUT TYPE=radio NAME=06sex value=’F’>Female<BR>
<INPUT TYPE=radio NAME=06sex value=’M’>Male
</BLOCKQUOTE>
<P>4. What is your age?
<INPUT TYPE=TEXT NAME=07Age SIZE=2 maxlength=3>
  years.<BR>
<P>5. What is the highest level of education you have completed? <BR>
<SELECT NAME=08Ed >
 <OPTION VALUE=’’ SELECTED>Choose from this list
 <OPTION VALUE=11> Less than 12 years
 <OPTION VALUE=12> Graduated High School (12 years education)
 <OPTION VALUE=14> Some College (13–15 years education)
 <OPTION VALUE=16> Graduated from College (Bachelor’s degree)
 <OPTION VALUE=18> Master’s degree
 <OPTION VALUE=19> Advanced Grad School beyond Master’s degree
 <OPTION VALUE=20> Doctoral Degree (Ph.D., M.D., J.D., etc.)
</SELECT><BR>
<P>6. Nationality (country of birth):
<INPUT TYPE=text NAME=09Cn SIZE=20 MAXLENGTH=30><BR>
<P>7. COMMENTS:<BR>
<TEXTAREA NAME=10CM ROWS=5 COLS=60 WRAP=virtual></
  TEXTAREA><BR>
<P>Please check your answers. When you are done, push the button
  below.
<P><INPUT TYPE=’submit’ VALUE=’finished’>
<H2>Thank You!</H2>
</FORM>
</FONT>
</BODY>
</HTML>
```

this case, the two questions are “In how many accidents have you been involved when you were the driver?” to be answered with a text box, and “I often feel fear when driving,” to be answered with a category rating scale. Table 27.2 shows a Web form created by SurveyWiz3, which illustrates text input, rating scales, pull-down selection list, and a text area for open-ended comments.

27.4.2 FactorWiz and WEXTOR

Birnbaum’s (2000b) FactorWiz program allows one to make within-subjects experimental factorial designs, with randomized order for the combinations. The program is even easier to use than SurveyWiz, once its concepts are understood. Information on how to use the program is contained in the works of Birnbaum (2000b, 2001a) and in the Web files containing the programs.

William Schmidt has written a freely available Perl script that works with SurveyWiz and FactorWiz. For instructions on how to install this generic perl script, see

http://ati-birnbaum.netfirms.com/Install_Perl_script.htm

Göritz and Birnbaum (2005) have contributed a free PHP program for facilitating data collection with HTML forms that also works with SurveyWiz and FactorWiz. It parses the input from any HTML form and automatically creates a MySQL database. See http://goeritz.net/brmic/.

Both SurveyWiz and FactorWiz create studies that are contained within a single Web page. To construct between-subjects conditions, one might use these programs to create the materials in the various conditions and use HTML pages with links to assign participants to different conditions. However, when there are a large number of within- and between-subjects factors, and when the materials should be put in separate pages so that response times and experimental dropouts can be traced, it becomes difficult to keep all of the files organized. In these cases, it is helpful to use WEXTOR (Reips and Neuhaus 2002), a program available on the Web at http://wextor.org that organizes experimental designs and keeps the various pages properly linked (see Section 27.5). For creating surveys, also consider http://surveys.deusto.es/, which presents a new, free, open source survey tool.

27.5 CREATING WEB EXPERIMENTS WITH COMPLEX DESIGNS: WEXTOR

Reips and his group have built several tools available from the *iScience Server* at http://iscience.eu that help Web experimenters in all stages of the experiment process: learning about the method, design and visualization, recruitment, and analysis of data. All of Reips' tools are Web based and therefore platform independent and can be used from any computer that is connected to the Internet. If you prefer a multiple-page survey with dropout measure, individualized random ordering of questions, and response time measurement, then you may want to use WEXTOR, a program that is reviewed in this section.

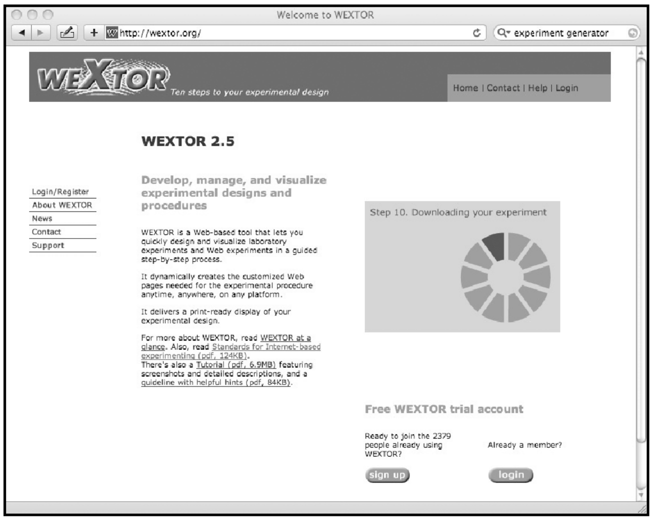
WEXTOR by Reips and Neuhaus (2002) is an Internet-based system to create and visualize experimental designs and procedures for experiments on the Web and in the laboratory. WEXTOR dynamically creates the customized Web pages needed for the experimental procedure. It supports complete and incomplete factorial designs with between-subjects, within-subjects, and quasi-experimental factors, as well as mixed designs. It implements server-side and client-side response time measurement and contains a content wizard for creating interactive materials, as well as dependent measures (graphical scales, multiple-choice items, etc.), on the experiment pages.

Many human factors considerations are built into WEXTOR, and it automatically prevents several methodological pitfalls in Internet-based research. This program uses nonobvious file naming, automatic avoidance of page number confounding, JavaScript test redirect functionality to minimize dropout, and randomized distribution of participants to experimental conditions. It also provides for optional assignment to levels of quasi-experimental factors, optional client-side response time measurement, randomly generated continuous user IDs for enhanced multiple submission control, and it automatically implements the meta tags described in Section 27.9. It also implements the warm-up technique for drop-out control (Reips 2000, 2002a), and provides for interactive creation of dependent measures and materials (created via content wizard).

The English version of WEXTOR is available at the following URL: http://wextor.org/wextor/en/.

Academic researchers can sign up free and can then use WEXTOR to design and manage experiments from anywhere on the Internet using a login/password combination. Support is available from tutorials, a FAQ, via a feedback and bug report form, a hints file, and an associated mailing list (e-group), all linked at the site. Figure 27.4 shows WEXTOR's entry page.

The process of creating an experimental design and procedure for an experiment with WEXTOR involves 10 steps. The first steps are decisions that an experimenter would make whether using WEXTOR or any other device for generating the experiment, such as listing the factors, levels, of within- and between-subjects factors, deciding what quasi-experimental factors (if any) to use, and specifying how assignment to conditions will function. WEXTOR produces an organized, pictorial representation of the experimental design and the Web pages and associated JavaScript and CSS files required to implement that design. One can then download the experimental materials in one compressed archive that contains all directories (folders), scripts, and Web pages.



WEXTOR 2.5

Develop, manage, and visualize experimental designs and procedures

WEXTOR is a Web-based tool that lets you quickly design and visualize laboratory experiments and Web experiments in a guided step-by-step process.

It dynamically creates the customized Web pages needed for the experimental procedure anytime, anywhere, on any platform.

It delivers a print-ready display of your experimental design.

For more about WEXTOR, read WEXTOR at a glance. Also, read Standards for Internet-based experimenting (pdf, 124KB).
There's also a Tutorial (pdf, 6.9MB) featuring screenshots and detailed descriptions, and a guideline with helpful hints (pdf, 84KB).

Free WEXTOR trial account

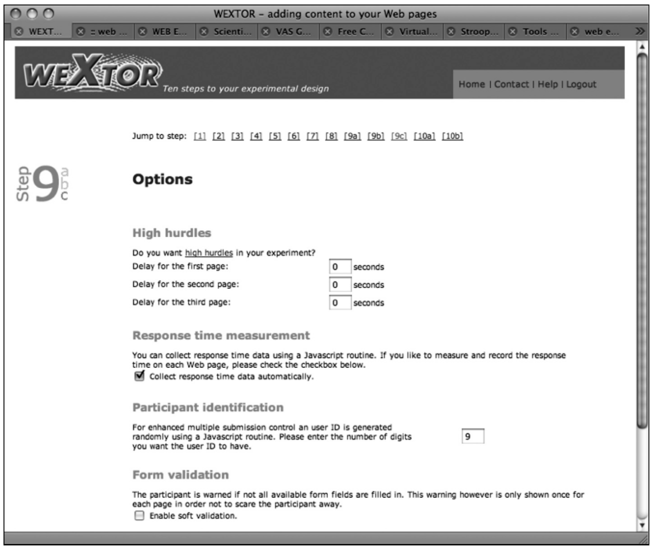
Ready to join the 2379 people already using WEXTOR?

Already a member?

FIGURE 27.4 **(See color insert following page xxx.)** The entry page to WEXTOR at http://wextor.org.

After decompressing the archive, the resulting Web pages created in WEXTOR can then be viewed (even when not connected to the Internet), tested, and further edited in an HTML editor. Afterward the whole folder with all experimental materials can be uploaded to a Web server. This can be done by FTP as described above for the case of an experimenter who does not operate the server, or it can be done by simply placing the files in the proper folder on the server, as described in the next section.

Since the first edition of this book, WEXTOR (now in version 2.5) has received a complete, usability-checked redesign and many features were added. Experiments can now optionally be hosted on the WEXTOR server, so an experimenter has the choice of using WEXTOR's free hosting service or of setting up his or her own server (see later in this chapter). WEXTOR now generates a code plan for each experiment. Many techniques for Internet-based experimenting were built into WEXTOR to automatically avoid common errors found in Internet-based research. The seriousness check technique, the multiple site entry technique, and the high hurdle technique (Reips 2000, 2002b, 2007) are now implemented. Any object can be integrated with the experimental materials, including for example, videos or Flash content (see Krantz and Williams 2010 for more information on using media in Internet-based research). Flexible timing out of individual pages and soft form validation can now be applied to the Web pages. Skins are available to flexibly change the appearance of all Web pages in the entire experiment at once. Figure 27.5 shows options for the implementation of the high hurdle


WEXTOR – adding content to your Web pages
WEXT...
:: web ...
WEB E...
Scienti...
VAS G...
Free C...
Virtual...
Stroop...
Tools ...
web e...
WEXTOR
Ten steps to your experimental design
Home | Contact | Help | Logout
Jump to step: [1] [2] [3] [4] [5] [6] [7] [8] [9a] [9b] [9c] [10a] [10b]
Step 9 a b c

Options

High hurdles

Do you want high hurdles in your experiment?

Delay for the first page: 0 seconds

Delay for the second page: 0 seconds

Delay for the third page: 0 seconds

Response time measurement

You can collect response time data using a Javascript routine. If you like to measure and record the response time on each Web page, please check the checkbox below.

☑ Collect response time data automatically.

Participant identification

For enhanced multiple submission control an user ID is generated randomly using a Javascript routine. Please enter the number of digits you want the user ID to have. 9

Form validation

The participant is warned if not all available form fields are filled in. This warning however is only shown once for each page in order not to scare the participant away.

☐ Enable soft validation.

FIGURE 27.5 Options for the implementation of high hurdle technique, response time measurement, session ID, and form validation in step 9 in WEXTOR.

technique (Reips 2000, 2002c), response time measurement, session ID, and form validation in step 9 in WEXTOR.

For experiments hosted at the WEXTOR Web site, data preparation can optionally be done on the server, so that data can then be downloaded ready for analysis in spreadsheet programs like Excel or SPSS. The downloaded data file contains a column showing the path taken by each participant (e.g., to see use of the back button) and *two* measures of response time for each accessed page (server side and client side).

Some research projects require dynamic tailoring of an experiment to the participant's sequence of responses. Table 27.3 shows a list of various tools and techniques used to create the materials for running experiments via the Web. In Web research, there are often a number of different ways to accomplish the same tasks. Table 27.3 provides information to compare the tools available and help determine which application or programming language is best suited for a given task.

Certain projects require a programming language such as CGI programming, JavaScript, Java, Flash, or Authorware programs. Programming power is usually required by experiments that rely on computations based on the participant's responses, randomized events, precise control and measurement of timing, or precise control of psychophysical stimuli. The use of JavaScript to control experiments is described by Birnbaum and Wakcher (2002), Birnbaum (2001a), and Baron and Siepmann (2000). JavaScript programs can be sent as source code in the same Web page that runs the study. This allows investigators to openly share and communicate their methods. That way, it becomes possible to review, criticize, and build on previous work.

Java is a relatively advanced programming language, and like JavaScript, it is intended to work the same for any browser on any computer and system. However, practice has shown that different implementations of the Java engine in different operating systems have not yet reached that goal. The use of Java to program cognitive psychology studies in which one can accurately control stimulus presentation timing and measure response times is described by Francis, Neath, and Suprenant (2000). Eichstaedt (2001) shows how to achieve very accurate response time measurement using Java.

Authorware is an expensive, but powerful application that allows one to accomplish many of the same tasks as one can do with Java, except it uses a graphical user interface in which the author can "program" processes and interactions with the participant by moving icons on a flow line. This approach has been used to good effect by McGraw, Tew, and Williams (2000; see also Williams, McGraw, and Tew 1999). However, participants are required to install a plug-in so this approach works best with participants in laboratories (where plug-ins can be preinstalled) or with online panels (where panelists will have installed them) rather than for experiments with open recruitment of participants at uncontrolled computers. Additional discussion of these approaches is given by Birnbaum (2000a, 2001a, 2004a, 2004b), Reips (2006, 2007), and Reips and Krantz (2010). Authorware was developed by Macromedia and acquired by Adobe; in 2010, Adobe announced that it will discontinue development of this program but will continue to support it.

JavaScript, Java, and Authorware experiments run client side. That means that the experiment runs on the participant's computer. This can be an advantage, in that it frees the server from making calculations and having a lot of traffic delays sending information back and forth. It can also be a disadvantage, if the participant does not have compatible script languages or plug-ins. At a time when Internet Explorer had a buggy version of JavaScript and JavaScript tended to produce error messages on people's computer screens, Schwarz and Reips (2001) found that JavaScript caused a higher rate of drop-out in Web studies compared with methods that did not require client-side programming. However, in recent years, JavaScript and Web browsers have become less error prone. Among the client-side programming options, JavaScript is probably the most widely used one.

There are certain tasks, such as random assignment to conditions, that can be done by HTML; by JavaScript, Java, and Authorware; or by server-side programs. By doing the computing on the server side, one guarantees that any user who can handle Web pages can complete the study (Schmidt 2000, 2007; Reips 2007). On the other hand, server side programs may introduce delays as the participant waits for a response from the server. When there are delays, some participants may think the program has frozen and may quit the study.

Server-side programs can analyze data as they come in and update a report of cumulative results. Perl and PHP are the two techniques most popular for such server-side programming.

The term "Ajax" (Garrett 2005) refers to a combination of techniques that allow continuous client-side updating with server-side information. AJAX stands for Asynchronous

TABLE 27.3
Purpose of Various Techniques and Tools to Create Materials for Web-based Research

Technique	Purpose	Pros	Cons/Considerations
HTML (basics)	To create Web content	Basic to everything on the Web	Takes a day or two to learn the most important tags
HTML Forms	To transmit data to and from participant	Basic device used in Web research	Takes a day to learn the basics
WYSIWYG Web Editors	To create Web content, including HTML forms	Help people who do not know HTML to make Web pages. Easy to make good-looking pages without understanding. Easy to learn.	Can create headaches for researchers. Students often fall into traps from which they cannot escape or may even remain unaware of fatal errors that spoil their project.
SurveyWiz	Creates one-page HTML surveys	Easy to learn and use. Creates surveys with text boxes and scales of radio buttons.	Limited in what it does. No detailed response time measurement.
FactorWiz	Creates one-page, within-subjects factorial designs with up to six factors	Easy to learn. Creates random orders of factorial material.	Same limits as SurveyWiz. Must be used repeatedly to create between-subjects conditions.
Web Survey Assistant	Creates a variety of surveys and CGI scripts	More powerful than SurveyWiz.	More difficult to learn than SurveyWiz.
WEXTOR	Creates multipage, between-subjects, within-subjects, quasi-experimental, or mixed factorial designs; based on HTML and JavaScript.	Easy to learn. Many human factors considerations built-in, prevents methodological pitfalls in Internet-based research.	Cannot do certain things that can be done with Java or Authorware.
Scientific LogAnalyzer	For analysis of log files (turns raw server data into files in “one participant per line” format)	Handles any type of log file. Includes module for dropout analysis. Calculates response times. Highly flexible.	One must ensure that the server saves the key information in the log file.
CGI Scripting (e.g., Perl, PHP)	Control the server, save data, security (e.g., passwords, online exams). Server-side programming.	Automation of many necessary processes such as saving data. Works for any browser.	Not easy to learn. Constant version updating necessary, due to security risks. May function more slowly than client-side programs.
JavaScript	Powerful programming language. Include dynamic interaction with participant. Control and measure time.	Client side, hence requires JavaScript turned on in the participant’s browser.	May not work properly in certain Web browsers.
Authorware	Construct experiments requiring timing, graphics, and dynamic interactions.	Products look good. Runs client side. Can do same tasks as Java but easier to learn. GUI.	Expensive. Plug-in download necessary on part of the participants.
Java	Powerful programming language, object-oriented.	Free. Usually preinstalled in browsers. Precompiled applets run on client side.	Very difficult to learn, may not work as expected with some Web browsers/ operating systems.

JavaScript And XML. These techniques allow one to create dynamic, interactive Web applications that can retrieve data from the server asynchronously in the background without interfering with the display and functional behavior of the existing page (Wikipedia 2010).

Technically, Ajax uses (1) a combination of HTML and cascading style sheets (CSS) for marking up and styling information, (2) the standardized Document Object Model (DOM) accessed with JavaScript to dynamically display and interact with the information presented, and (3) a method for exchanging data asynchronously between browser and server, thereby avoiding page reloads. Thus, on a Web page created

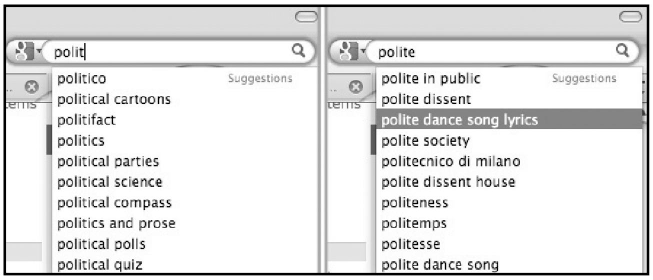


FIGURE 27.6 An example of Ajax scripting: Autocomplete. The list of suggestions changes with each letter that is typed, here from (left) “polit” to (right) “polite.”

with Ajax technologies a visitor will see responses to his or her actions without having to wait for pages to reload.

An example can be seen in Figure 27.6, which shows an *autocomplete script* at work. The Web form provides a list of likely options that adapts to each new letter that is typed in the search field. Users not only can save time, because they don't need to fully type long options, but autocomplete also supports retrieval from memory. Autocomplete is a good human–computer interface implementation of a finding from memory research; namely, recognition is easier than recall. Of course, there is always the danger that autocomplete will cause the user to make errors by posing an unintended response.

Although some tasks can be done on either client or server side, there are certain tasks that can or should only be done by the server, such as saving data or handling issues of security (e.g., passwords, using an exam key to score an IQ test or an academic exam, etc.). Of course, to program the server, one needs to have access to the server.

27.6 RUNNING YOUR OWN SERVER

Running a Web server and making documents available on the Web has become increasingly easy over the years as manufacturers of operating systems have responded to demand for these services. Even if there is no preinstalled Web server on your system, installing one is neither complicated nor expensive. Thanks to contributions by an active "open source" community, there are free servers that are reliable and secure, along with free programming languages such as Perl and PHP that allow one considerable power for running and managing research from the server side (Schmidt 2000).

The free Apache Web server, available for a wide range of operating systems, can be downloaded from http://www.apache.org/. On this Web site there is also up-to-date documentation of the details of the installment process. Anja Göritz has provided tutorials on installation of the free Apache server along with PHP and MySQL, see

http://www.goeritz.net/ati/

Running your own Web server (rather than depending on your institution) confers several advantages (Schmidt, Hoffman, and MacDonald 1997). First of all, you can have physical access to the server, allowing you to directly observe and control it. You can, for example, easily add and delete files to your Web site by moving files from folder to folder; you can disconnect the server, modify it, and restart it.

Second, Web servers of institutions are restricted because they have to fulfill many tasks for different purposes. Consequently, many settings are designed to satisfy as many requirements as possible (one of which is reducing the likelihood of getting the network supervisors into trouble). On your own server, you can install and configure software according to the requirements of your own research (for example, you should change the server's log file format to include the information mentioned in Section 27.7).

Third, you can have greater harm on your access to the server if you operate it yourself. The institution's administrators do in terms of hindering research may outweigh any help they might provide with technical issues. You can try to explain your research goals and provide assurances that you would not do anything to weaken the security of the system or to compromise confidential files. Still, some administrators will resist efforts by researchers to add CGI files that save data to the server, for example, fearing that by error or intent, you might compromise the security of the system. Some Institutional Review Boards (IRBs) insist that administrators deny researchers complete access to the server for fear of compromising privacy and security of data collected by other researchers who use the same server.

Even if you run your own server within an institution you may run into trouble regarding the domain names and IP addresses by which your server and the services on it are accessed. The institution might decide to change or reassign names or delete them. To avoid such problems it is best to register your own domain name (or names) with an independent provider and announce that name to your participants and the research community. In case something goes wrong at your institution you will be able to quickly point the domain name at a different IP address.

In the example below, we will describe how to create and publish Web pages for Internet-based studies using Apple's Macintosh OS X operating system (see Wikibooks n.d.). Things work similarly on PC, but may vary depending on the exact version of the Windows or Unix/Linux operating system and its particular security settings (see Göritz 2006).

27.6.1 Place Your Materials in the Designated Folder

In your private area ("Home") under Mac OS X there is a folder called "Sites" (or "Websites"). Put the folder with your materials (in this example the folder is named, "my_experiment") in the "Sites" folder. Figure 27.7 shows the respective Finder window. No other files need to be installed, if you created your experiment with SurveyWiz, FactorWiz, or WEXTOR. In case you are an advanced user of Internet technology and you would like to use Perl or PHP scripts for database-driven Web studies, you need to configure the system accordingly.

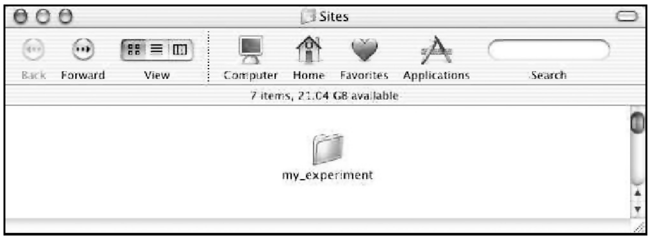


FIGURE 27.7 Web pages go in the Sites folder (or "Websites," in some versions of the operating system), in this case a folder with experimental materials created in WEXTOR.

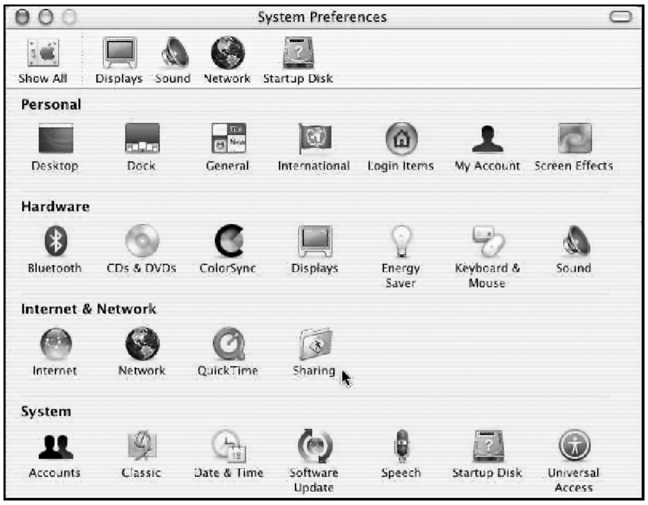


FIGURE 27.8 (See color insert following page xxx.) The second of three clicks to turn on the Mac Server is to click on "Sharing" in the System preferences.

On the Macintosh, Perl, PHP, and mySQL are built-in, but they need to be configured using procedures described in more detail in Section 27.6.4, and in the Web site that accompanies this chapter. For Windows PCs, see the tutorial by Göritz (2006).

27.6.2 Turning on the Web Server in Mac OS X

The Apache Web Server comes already installed on new Macintosh computers. Turning on the Web server under Mac OS X takes three mouse clicks: First you need to open the System Preferences (Click I), then click on "Sharing" (Click II, see Figure 27.8), and then click on "Personal Web Sharing" (Click III, the "Services" tab will be preselected), as shown in Figure 27.9.

Before you can actually make anything available on the Web using the built-in Apache Web server you need to make sure that your computer is connected to the Internet. However, you can always test your site by "serving" the pages to yourself locally, i.e., view them as if you were surfing in from the Web. Here is how you do this:

- Open a Web browser
- Type "http://localhost/~USERNAME/my_experiment/" into the browser's location window (where USERNAME is your login name).

The exact address will actually be shown at the bottom of the system preferences window displayed in Figure 27.9 (not shown here to preserve privacy).

27.6.3 Where to Find the Log Files

The default storage for the log files created by the Apache server that comes with Mac OS X 10.5 or newer is /var/log/apache2/access_log (or, in earlier versions of the operating system: /var/log/httpd/access_log). It is a text file readable in any text editor. A nice freeware application to read log files is LogMaster (check on http://www.versiontracker.com for download). You can directly copy the log file and upload it to Scientific LogAnalyzer at http://sclog.eu (Reips and Stieger 2004) for further analysis (see next section).

To view the log file in Mac OS X, you would open the *Applications* folder, open the *Utilities* folder and double click the *Terminal* application. A terminal window will open up. This window accepts old-fashioned line commands, and like old-fashioned programming, this Unix-like terminal is not forgiving of small details, like spaces, capitalization, and spelling.

Carefully type the following command:

```
open /var/log/apache2/access_log
```

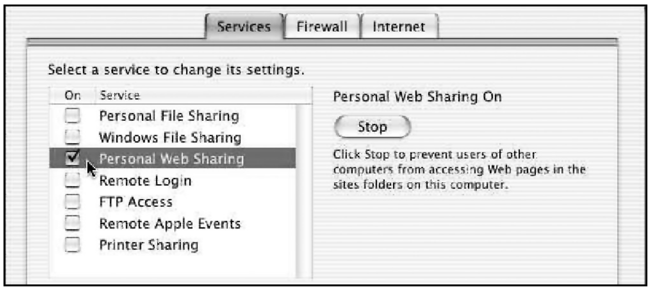
(or, on earlier versions of the operating system: `open /var/log/httpd/access_log`)

Before you hit the return key, look at what you have typed and make sure that everything is exactly correct, including capitalization (here nothing is capitalized) and spacing (here there is a space after the word "open"). If you have made no typos, when you press the return key, a window will open showing the log file for the server.

The default logging format used by Apache is somewhat abbreviated. There is a lot of useful information available in the HTTP protocol that is important for behavioral researchers (see Section 27.7) that can be accessed by changing the log format, for example to include information about the referring Web page and the user's type of operating system and Web browser. Methods for making these changes to the configuration of Apache Server are given in the Web site that accompanies this chapter and in the FAQ at http://wextor.org.

If you created your study with WEXTOR, you are ready to collect data.

If you used SurveyWiz or FactorWiz, and if you want to save data to your own server, rather than download it from the *psych.fullerton.edu* server, you will need to make the adjustments in your HTML page(s) and server. There are two approaches. One is to use a CGI script to save the data, installing a CGI on your own server to replace the generic PolyForm script provided by Birnbaum (2000b); this technique will be described in Section 27.6.4. The other approach



Select a service to change its settings.

On	Service
□	Personal File Sharing
□	Windows File Sharing
☑	Personal Web Sharing
□	Remote Login
□	FTP Access
□	Remote Apple Events
□	Printer Sharing

Personal Web Sharing On

Click Stop to prevent users of other computers from accessing Web pages in the Sites folders on this computer.

FIGURE 27.9 The third click turns on Web sharing in the Web server in Mac OS X.

is to send the data by “METHOD=GET” to the server’s log file.

To use the “GET” method to send data to the log file, find the <FORM> tag of your survey:

```
<FORM METHOD=’post’ ACTION=’http://
psych.fullerton.edu/cgi-win/polyform
.exe/generic’>
```

and change it to

```
<FORM METHOD=’GET’ ACTION=’[address of
next Web page here]’>.
```

Where it says “[address of next Web page here]” you need to specify the name of a Web page on your server, for example, ACTION=ThankYou.htm (but it could even be to the same page). It is conventional to use a page where you thank your participants and provide them with information such as how to contact you for any comments or questions about your study. In Section 27.7 you will be shown how to record and process the form information that is written to the log file each time someone participates in your study.

27.6.4 Installing a Perl Script to Save Data via CGI

The procedures for installing a Perl script with the Apache server for Windows PC are described by Schmidt (2003). This section explains how to use Perl on the Mac OS X.

First, create a folder to hold the data files. Open the icon for the Macintosh hard drive. From the File menu, select *New Folder*. A new folder will appear, which you should name *DataFiles*. Now click the folder once, and press the Apple Key and the letter “i” at the same time, which opens the “Get Info” display. Then click on the pop-up list and choose *Privileges*. Set the first two privileges to *Read and Write* and the third to *Write Only (Dropbox)*.

Next, examine the Perl script in Table 27.4. This is a CGI script, written by Schmidt to emulate the generic PolyForm script used by Birnbaum (2000b, 2001a). This script will save data from a Web form to a new data file in the above folder. You can edit the first three lines to suit your own configuration and needs. For example, you can change the URL in the second line to the address of a “thank you” page on your own server, or you can change the location on your computer where you wish to save the data by changing the third line.

The Perl script in Table 27.4 is also available from the Web site for this chapter, which will save you from typing. You should save this as a Unix text file, with an extension of “.pl” (where “l” is a letter, not a number). For example, you can save it as *save_data.pl*. This file should be saved in the following folder. From your Mac hard drive, open the *Library* folder, then open the *WebServer* folder, and then open *CGI-Executables*. Place *save_data.pl* in this folder.

Now you need to open the Terminal. Open the *Applications* folder and within it open the *Utilities* folder. Double click

TABLE 27.4
CGI Script in Perl that Works with SurveyWiz and FactorWiz.#

```
#!/usr/bin/perl
$redirect_to = “http://psych.fullerton.edu/mbirnbaum/decisions/thanks
  .htm”;
$path_to_datafile = “/DataFiles/”;
use CGI;

$query = new CGI;
#timestamp the submission
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) =
  localtime(time);
$mon++;

#determine the data filename, open and dump data
$filename = $query->param(‘00exp’);
open(INFO, “>>$path_to_datafile/$filename.data”);

foreach $key (sort($query->param))
   {
   $value = $query->param($key);


   #filter out “’s and ,’s
   $value =~ s/\”/\’/g;
   $value =~ s/,/ /g;
   if ($value !~ /^pf/)
   {
      print INFO “\”$value\”, “;
   }
   else
   {
      # filter out items that need to be expanded at submission time pf*
      if ($value =~ /^pfDate/)
      {
         print INFO “\”$mon/$mday/$year\”, “;
      }
      if ($value =~ /^pfTime/)
      {
         print INFO “\”$hour:$min:$sec\”, “;
      }
      if ($value =~ /^pfRemote/)
      {
         print INFO “\””,$query->remote_addr(),”\”, “;
      }
      if ($value =~ /^pfReferer/)
      {
         print INFO “\””,$query->referer(),”\”, “;
      }
   }

   #print “$key:$value”;
  }
print INFO “\”complete\”\n”;
close (INFO);

print $query->redirect($redirect_to);
exit();
```

the *Terminal* program and it will open. Type the following command:

```
chmod ugo+rwx /Library/WebServer/CGI-Executables/save_data.pl
```

Before you hit Return, study what you have typed. Be sure that the capitalization, spacing, and spelling are exactly correct. There should be a space after "chmod" and before "/Library". Hit Return, and a new prompt (new line) appears.

Now, take the survey made by SurveyWiz or FactorWiz (the Driving Survey example in Table 27.2 will do), and find the `<FORM>` tag. Change this tag to read as follows:

```
<FORM METHOD='post' ACTION='http://
localhost/cgi-bin/save_data.pl>
```

Save your survey in the folder "Sites" as described in Section 27.6.1. You are now ready to collect data. Use your browser to view the survey. Fill it out and push the "submit" button. You should be redirected to the "thank you" page designated on the second line of the Perl Script, and your data should have been saved in the folder, *DataFiles*.

With this one CGI script, you can do all of the experiments and surveys described by Birnbaum (2001a), without needing to install another script. The data will arrive in the designated folder on your server as comma separated values (CSV) files that can be easily imported to many statistical and spreadsheet applications. Methods for working with and analyzing data collected in this format are described in detail by Birnbaum (2001a).

27.7 HOW TO ANALYZE LOG FILES

Scientific investigation relies on the principle of raw data preservation. Raw data need to be saved for scrutiny by other researchers from the community (American Psychological Association 2010), for example to aid in reanalysis or meta analysis. This principle applies to Internet-based research as well, where it can be argued that server log files, properly configured, are the raw data (Reips 2001, 2002b, 2007; Reips and Stieger 2004). In addition to all data sets from full participation, complete log files from a Web server used in an Internet-based investigation contain the following useful information:

1. Data about potential participants who decide not to participate (e.g., the number of people who see the link to a study, but don't click on it).
2. Data about technical conditions during the investigation (i.e., the general Web traffic conditions at the server and the particular conditions of each request by a participant).
3. Data sets for those who provided incomplete information; partial nonresponse may reveal information about potential problems (Reips 2002b).

Not reporting the information listed above, for example because one did not save it, carries the great danger of misleading scientists with respect to the main effects of variables (Birnbaum and Mellers 1989; Reips 2000, 2002c; Reips and Stieger 2004). Even when drop-out rates are equal in two groups, the observed trend may easily show the opposite of the true effect that one would have found had all data been complete. Because participants find it quite easy to quit Web studies, there can be sizeable attrition in such studies, and this attrition needs to be saved and reported (Frick, Bächtiger, and Reips 2001; Knapp and Heidingsfelder 2001; Reips 2002b, 2007).

Log files need to be configured so that the information needed will be saved. Often the log file format follows a common predefined format, for example the Webstar log file format. This log file format is used by the Web Experimental Psychology Lab and contains information in the following order: CONNECTION_ID DATE TIME RESULT HOSTNAME URL BYTES_SENT AGENT REFERER TRANSFER_TIME SEARCH_ARGS.

To change the log file format on your Web server do the following. On Mac OS X open the Terminal application as described above. Then evoke the pico text editor to edit a file named "httpd.conf." Type

```
sudo nano /private/etc/apache2/httpd.conf
```

(If this doesn't work, i.e., you can't find the location of httpd.conf, then type httpd –V and get the location from the list that will appear.)

Scroll down or search for lines beginning with "LogFormat". Below the block of lines you will create a new line by hitting "Return". Then type (or copy from the accompanying Web site or http://wextor.org/wextor/en/faq.php):

```
LogFormat "%P\t%{%m/%d/%y}t\t%{%T}t\t%X\t%h\
    t%v%U\t%B\t%{User-Agent}i\t%{Referer}i\t%T\
    t%q\t%D" my_logfile
```

This directive lists a sequence of field identifiers, each of which tells the server to log a particular piece of information. Fields are separated by tab characters, indicated by "\t".

Next, do a search for "CustomLog." It specifies the location to which the log file will be written. You should see a line similar to:

```
#CustomLog "var/log/apache2/access_log" common
Add the following line below it:
CustomLog "var/log/apache2/access_log" my_logfile
```

Once you have made the changes, save the document (if in the pico text editor) by pressing the control and "O" keys. You will be prompted to confirm the save. Just press return to do so. Finally, quit pico by pressing the control and "X" keys and restart Apache:

```
apachectl -k graceful
```

In the format you configured with the LogFormat command each string means the following:

\ : Wildcard used for separating the strings in cases where spaces, ‘%’, or ‘}’ are not used.

t : Tabulator,

%P : The ID of the process that serviced the request.

%{%m/%d/%y} : The date, here in the format Month/Day/Year,

%{%T}: The time, here in the format HH:mm:ss,

%X : Connection status when response is completed (X = connection aborted before the response completed, + = connection may be kept alive after the response is sent, – = connection will be closed after the response is sent),

%h : Remote host (client’s computer IP)

%v : The canonical ServerName of the server serving the request.

%U : The URL path requested, not including any query string.

%B : Size of response in bytes, excluding HTTP headers.

%{User-Agent}i : User-Agent information from the header (web browser, operating system).

%{Referer}i : Referer information from the header.

%T : The time taken to serve the request, in seconds.

%q : The query string (prepended with a ? if a query string exists, otherwise an empty string)

%D : The time taken to serve the request, in microseconds.

The following shows an example of a line in the log file, in this case of accessing the Web Experimental Psychology Lab at http://www.wexlab.eu/.

```
3169 02/25/10 13:43:37 + 130.206.135.23 www.wexlab.eu/ 253 Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; es-ES; rv:1.9.2) Gecko/20100115 Firefox/3.6 - 0 775
```

A platform-independent interactive Web site that helps Internet researchers in analyzing log files is Scientific LogAnalyzer (Reips and Stieger 2004, http://sclog.eu). It was created to meet the needs of those who collect data on the Internet. Scientific LogAnalyzer provides an option for selecting the predefined format mentioned above, and it also contains procedures to identify and process any type of log file. To match a predefined format, the user may also rearrange columns in the log file before uploading it to Scientific LogAnalyzer, which can be done easily in text editors and spreadsheet programs.

Scientific LogAnalyzer has features important to behavioral and social scientists, such as handling of factorial designs, response time analysis, and dropout analysis. Scientific LogAnalyzer was developed to include calculation of response times, flagging of potential multiple submissions, selecting either first or last response from same IP, marking of predefined IP addresses and/or domain names, and free definition of session timeout). The program is highly flexible on the input side (unlimited types of log file formats), while strictly keeping the traditional one-case-per-row output format. Other features include (1) free definition of log file format, (2) searching and identifying any combination of strings (necessary for organizing conditions in experiment data), (3) computation of approximate response times, (4) a module for analyzing and visualizing dropout, (5) detection of multiple submissions, (6) output in HTML and/or tab-delimited files, suited for import into statistics software, (7) speedy analysis of large log files, and (8) extensive help from an online manual.

27.8 PRETESTING OF INTERNET-BASED RESEARCH

Before an experiment is placed on the Web, it is necessary to perform a number of checks to make sure that the study will yield useful data. First, one should be clear on how the data will be analyzed and that the study will answer the question it is supposed to answer. This check is basic and applies to laboratory research as well as Web research. Those who are unclear on how the data will be analyzed almost never devise a study that can be analyzed.

Second, one should conduct checks of the HTML and CGI script to code and save data to ensure that every possible response is properly coded and recorded in the proper place in the data file. One should check that every radio button (in a given item) functions properly and that answers to one question do not overwrite responses to another item. This is one of the advantages of using a Web service like FactorWiz (Birnbaum 2000b) or WEXTOR (Reips and Neuhaus 2002); these programs save time by automatically creating safe code.

Third, one should test some participants in the laboratory. Observe them as they read and respond to the materials. Ask them to identify aspects of instructions that are unclear. With a few exceptions, Internet-based research does not have a laboratory assistant who can answer questions, so every question must be addressed in advance. Check if people are responding before they have scrolled to see the information that they are supposed to review before responding. One of the second author’s students had placed a response device before the material the participant needed to read. During pilot testing, the second author observed a number of people who responded before they had scrolled to make visible what they were supposed to be judging. Analyze the data from the pilot study to see that the coding and analysis will function properly. It is often when analyzing data that students discover problems with their studies. That is why some pilot data should be analyzed before the main study is run.

One can discover a lot by observing participants in pilot research. For example, in a study with the random response method (e.g., Musch, Bröder, and Klauer 2001), participants were supposed to toss two coins and then respond “yes” if both coins were heads, “no” if both coins were tails, and to tell the truth otherwise. The purpose of the technique is to allow an experimenter to assess a population statistic without knowing any person’s true answer. For example, most people would be embarrassed to admit that they cheated on their income taxes, but with the random response method there is no way to know if “yes” meant that the person did cheat or that the coins were both heads. If people follow instructions, this method allows the experimenter to subtract 25% "yes" answers (that occurred because of two heads) and 25% "no" (resulting from two tails), and double the remainder to find the correct proportions. For example, if 30% of the group indicated that they cheated on their taxes, it means that 10% of the population cheated. In our pilot test, however, only one of fifteen participants took out any coins, and she asked first if she should actually follow the instructions. The second author urged his student to add stronger instructions and an extra pair of items at the end of the survey asking if the participant had actually used the coins, and if not, why not. About half said they had not followed the instructions, giving excuses such as "lazy" and "I had nothing to hide." This example should make one very concerned about what happens in studies that are not observed, and still more concerned about studies that are launched without pilot testing.

It is also important to pretest the materials with different browsers and systems and with a small monitor, to make sure that everyone will see what they are supposed to see. Otherwise, you would probably run into what the first author coined "configuration error IV" in Internet-based research—the all too common underestimation of the technical variance inherent in the Internet (Reips 2002b; also see Reips 2007; Schmidt 2007). Consider adding items to your research instrument to ask about monitor sizes, volume settings, and other settings that you think might make a difference to the results. Such information can be used to partition the sample for separate analyses. Considerations of delivery of psychophysical stimuli and a discussion of when such settings may or may not matter are reviewed by Krantz (2001) and Krantz and Williams (2010).

27.9 RECRUITING PARTICIPANTS FOR INTERNET-BASED RESEARCH

Participants can be recruited by traditional means, such as course assignments, student "participant" pools, face-to-face requests, word of mouth, posters, flyers, newspaper advertisements, etc. We will focus here on Internet methods such as recruitment via Web site, mailing list, online panel, newsgroup, e-mail, listings, blog, and banner ads. Recruitment for Internet-based studies can be made much more effective by using one or several of the techniques described by Birnbaum (2001a) and by Reips (2000, 2002b, 2002d, 2007).

27.9.1 Recruitment via Web Sites and Search Engines

One natural way to recruit participants is via one's own home page. However, many personal home pages are rarely visited. Besides, visitors of your home page may know too much about your research (or read about it on the page) to be suitable (i.e., naïve) participants for certain studies. (A useful strategy to test whether such self-selection may be biasing your results is the *multiple site entry technique* (Reips 2000, in press), which will be described in Section 27.9.2.4.)

An institution's Home page, for example a university's, may be a better choice for recruiting large numbers of participants. In addition, an institution's Home page will often convey a more legitimate impression than a personal home page. However, it will not be easy to get agreement to announce your study on an institution's Home page unless a number of administrators have agreed that it is acceptable to do so.

Some of the best places for recruitment are institutionalized Web sites for Internet-based experimenting, such as the Web experiment list (http://wexlist.net), the Web Experimental Psychology Lab (http://wexlab.eu), and the Psychological Research on the Net list (http://psych.hanover.edu/research/exponnet.html) by John Krantz, who published some of the first Web experiments (Krantz, Ballard, and Scher 1997; Welch and Krantz 1996). Some of these Web sites are visited by thousands of potential participants every month (Reips 2001a, 2001b, 2007; Reips and Lengler 2005), and some managers even provide you with a free check of your experiment, before linking it. A link on a Web experiment site may also serve an archiving function, as an example for future studies, and as a reference in publications.

People coming to your study via one of these Web research sites are true volunteers who have already decided that they want to take part in one or more psychology studies, and who chose your study from a list of ways to participate. So, the concerns one might have with respect to students who are participating only to fulfill an assignment are relieved with this source of participants.

To recruit participants by means of search engines, you can enhance your pages to help people find your study. Suppose you wanted to recruit people interested in psychology. To help those who are looking on the Web for "psychology," you can put that word in the title of your site, add it to the early text in your site, and add meta tags including "psychology" as a key word.

Suppose you wanted to recruit people with rare characteristics, such as transvestites or people who suffer from sexsomnia (for a study, see Mangan and Reips 2007). You could include meta tags in your entry page in order to help these people find your study. Table 27.5 shows an example of a Web page that might be used to recruit participants to a survey of memories of early experiences by transvestites. You should

TABLE 27.5
Use of Meta Tags to Recruit via Search Engine

```
<HTML>
<HEAD>
<META NAME="keywords" CONTENT="transvestites,cross-
  dressing,survey,early experiences,psychology,research">
<META NAME="description" CONTENT="research psychologists
  invite transvestites to complete a survey of early life experiences that we
  hope will contribute to understanding this condition">
<TITLE>Survey of Early Experiences of Transvestites</TITLE>
</HEAD>
<BODY>
(Further information on the study and a link to the study would be placed
  here)
</BODY>
</HTML>
```

also consider using an informative title for the first page of your study. Reips (2002b) advises using uninformative page titles and page names for consecutive pages to avoid the possibility that people will find these subsequent pages via search engines and enter the study somewhere in the middle. However, the first page that recruits participants can have an informative title without distracting the participant.

Meta tags can (and should) also be used to keep search engines *away* from all pages except the first page (you may even desire to keep search engines away from that page, if you would like to recruit via other means exclusively). The "robots" tag in Table 27.6 needs to be set to "none," because the routines used by search engines to search the Web for new Web pages are called "robots" (and sometimes "spiders" and "crawlers") and this technique informs them there is nothing for them.

A third important task that can be handled by meta tags is to prevent caches in search engines, mediating servers, and proxy servers from serving old versions of research materials after they have been updated. Caches contain stored files downloaded from the Web that are stored for reuse later. For example, your Web browser may be configured to store HTML code, images, and other media from pages you visited in its own cache. Later, when you request that page again, the Web browser quickly checks in the cache if it holds any of the text and media (e.g., images) you are requesting and displays them to you. This way, the page can be displayed more quickly (than if you had to wait for all the same files to download again), and much unnecessary traffic is avoided.

However, the material loaded from the cache may be outdated: If an experimenter finds an error in her Internet-based study and replaced the Web page on her server, users may continue to see and even be served the old version. Why? Because of caches holding the old version in proxy servers. Proxy servers hold huge caches, but not only for one computer—they hold all of the Web traffic going into and coming out of entire institutions. As you can imagine, there are some interesting analyses that can be performed with data collected on proxy servers (Berker 1999, 2002). Table 27.6 shows two meta tags below the robots tag that will keep proxy servers from caching your Web pages. Web experiments created with Version 2.1 or newer of WEXTOR automatically contain the meta tags described in this section.

TABLE 27.6
Use of Meta Tags in Pages after the Entry Page

```
<HTML>
<HEAD>
<meta name="AUTHOR" content="WEXTOR">
<meta name="ROBOTS" content="NONE">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="expires" content="Thursday, 1-Jan-1991 01:01:01 GMT">

<TITLE> </TITLE>
</HEAD>
(the body of the Web page goes here)
```

27.9.2 Recruitment via Other Means

In a survey of the "pioneers" of Web experimenting, Musch and Reips (2000) asked the question "In which media did you announce your experiment?" A surprisingly large number of Web experiments were not announced solely on the Web. Researchers also used newsgroups (18 of 35 experiments), e-mails (15), and search engines (14) to advertise their Web experiments. Only very few researchers used radio (1) or print media (2), although we now know that these offline media can be extremely effective. More recently, it has become possible to quickly recruit many participants via online media. The first author and his students have repeatedly announced Web experiments in http://20min.ch, the online version of "20 minutes," Switzerland's newspaper with the largest circulation nationally. Within a few days, each of the experiments was visited by several thousand participants.

Birnbaum (2001a) gives a number of suggestions regarding recruitment. He recommends against sending e-mail messages to people who do not want to get e-mail from you for reasons we will describe in Section 27.9.3 on ethics below. A better method for recruiting from a listserv is to contact the organization that runs the listserv and ask this organization for help. If you can convince the organization that your research is serious and will be of value to the community that the organization serves, you can often get good help from them. They can post an announcement in their Web site, send invitations in their newsletter, and even post messages for you to their members via e-mail. For example, to recruit transvestites, one could contact organizations to which these people belong, show that your research would be of interest or benefit to the members, and ask the organization to recruit for you.

This suggestion was adopted by Drake (2001), a graduate student who was working with Birnbaum at the time. She wanted to recruit elderly people with interests in genealogy. She realized that members of her target group belong to organizations that have an Internet presence. Drake contacted an organization, which saw the potential interest and value of her research to their members and agreed to send an e-mail to the members on her behalf, vouching for her research and inviting participation. Within one week, she had more than 4000 completed data records and many messages of encouragement and support (Drake 2001).

27.9.2.1 Newsgroups

In the early days of the Internet, communications consisted largely of text-based materials. Many people had very slow connections and logged onto virtual *bulletin boards* via modem to upload and download messages. One of the largest bulletin board networks was USEnet. Most of the USEnet newsgroups were saved and made accessible on the Web by deja.com (playing on the term *déjà vu*), a service that was later acquired by Google. Google now lists all available newsgroups as *Google groups* (http://groups.google.com/) and makes them available for searches. For example, if you surf to Google groups and type in the words "Web experiment survey" the two first postings listed are those originally posted to the newsgroup "sci.psychology.research" by Musch and Reips, inviting Web experimenters to their survey mentioned above.

If you use newsgroups for recruitment of participants, then be aware of the following points:

- Your posting will remain indefinitely, until the newsgroup might be erased from the Internet (if the study has an ending date, you should include that in your post).
- It will take several days until your posting is widely available.
- Make sure your posting is considered appropriate in the newsgroups to which you post—ask the moderator first, if the newsgroup is moderated.

27.9.2.2 Participant Pools and Panels

Many colleges have established pools of students who volunteer to participate in studies. In some cases, the researcher can request a "stratified" sample of participants, stratified with respect to gender, for example. Such a traditional participant pool may be an effective way for recruitment in Web studies, especially at large colleges.

However, student samples are not stratified with respect to age or education, nor are students heterogeneous with respect to many other variables (Birnbaum 2001a; Reips 2000). Most are between 18 and 22 years of age; all are graduates of high school, and none are graduates of college. Psychology pools now contain about two-thirds females, and at many schools, students who take psychology are more likely than the typical student to be undecided majors. Birnbaum (1999b) wanted to study correlates of education (especially education in decision making) and therefore needed to recruit off campus to obtain a sample that would show wide variation in education.

Using the Internet, a number of academic researchers and market researchers have created participant pools of people with more heterogeneous characteristics than college students by means of the Internet (Baron and Siepmann 2000; Smith and Leigh 1997; Göritz 2007, 2009). If nurtured properly, these "online panels" or "online research panels" of volunteers for Web-research are a means of widely distributing easy access to a wide range of participants.

27.9.2.3 E-mail

One very effective way of recruiting participants involves bending e-mails to mailing lists of people who want to receive your mail. At a conference (SPUDM, Zürich 2003), the first author heard an interesting paper on the first day of the conference and decided to replicate that study overnight. He was able to include the results in his talk on the second day of the conference, in order to demonstrate how efficient Web-based research can be in comparison with the methods used by the original authors (see link to Reip's SPUDM presentation on companion Web site at http://wexlab.eu/hcihandb). Within 8 hours, complete data sets from 162 participants (compared to 64 in the original study) were recorded in the Web experiment, most of which were recruited via three mailing lists.

27.9.2.4 Banner Ads

People who visit tattoo and piercing sites may have different personalities than those who visit traditional family values sites. Buchanan (2000, 2001) has exploited this insight to form criterion groups from people who are recruited by different methods, in order to validate personality tests. He has had some success comparing people recruited from different user groups. This technique is sometimes called the *multiple site entry technique* (Reips 2000, 2002b), which involves recruiting people via several different methods or sources and comparing data between these groups. Buchanan (pers. comm., Oct. 10, 2002) reported, however, that response to banner ads has been very meager and probably not worth the money (for similar results, see Tuten, Bosnjak, and Bandilla 2000). Because banners usually represent commercial advertising, a banner ad for a scientific study is hard to distinguish from a deceptive come-on for a commercial message.

27.9.2.5 Password Techniques

Passwords can be used to control entry to a study or survey, or they can be used to validate data from members of a voting body. Passwords are one way to determine if one person is trying to "stuff the ballot box" with multiple submissions of the same vote.

Password techniques (Reips 2000; Schmidt 2000) can be used to guarantee authenticity and originality of a participant's identity. In an experimental study on Internet-based versus paper-and-pencil-based surveying of employees, Reips and Franek (2004) printed anonymous individual codes for use in the (mailed) invitation to 655 employees. From the log analysis of the Internet-based data, the following could be determined:

- Whether any codes were used repeatedly (showing multiple submissions, and permitting the investigators to include the first set only).
- Whether people without code tried to access the survey (using wrong codes or no codes).

There are many other methods for authenticating data (see Schultz 2011) and detecting or preventing multiple submissions of data. These are summarized in a number of papers, including Birnbaum (2004a, Table 1).

27.9.3 Ethics and Etiquette in Internet Science

It is difficult to injure someone in research via the Web, except by dishonesty, so the fundamental ethical principle for Web-based research is honesty. If you promise some benefit for participation, then you must follow through and provide that benefit. For example, if you promise to pay participants, then you must pay them. Failure to do so is fraud. If you promise to give participants a report of the results by a certain date, then you must provide them the report. A good way to do this, incidentally, is to post the results to the Web and send the URL to those who requested the results (but do not send attachments, see below). If you promise to provide a chance in a lottery to win a prize, then you must follow through and run the lottery and distribute the prizes that were promised.

27.9.3.1 Taking Precautions

If participation is not anonymous, and if you promise confidentiality, then you must do everything you can to secure the data and prevent them from becoming public. Birnbaum (2001a) lists a number of steps to provide security of the data on the server. Keep the door to the server locked, and do not leave keys or passwords around. Record only what is necessary, keep personal information separate from other data, if possible, and remove identifying information from stored data as soon as possible. The server should be kept up to date with respect to security from hackers.

Although Musch and Reips (2000) reported that early Web experimenters had not found hackers to be a problem, the potential is certainly there for an unfortunate incident. Reips (2002b) observed a high rate of insecure servers in Web studies ("configuration error I"). During routine checks of Web experiments by researchers who applied for inclusion in the Web Experimental Psychology Lab, Reips was able to download openly accessible data files in several cases. Apparently, many researchers were not aware of the vulnerabilities of their Web servers and operating systems, and were surprised when he sent portions of their confidential data files to them. On the basis of statistics permanently collected by http://www.securityfocus.com ("bugtraq"), Reips (2002b) also noted the better track record of the Mac OS against vulnerabilities to hackers and viruses, compared to Windows and other operating systems. Depending on the operating system used, an Internet scientist will have to invest less or more time in keeping up with newly discovered vulnerabilities and respective security developments. Open source software is often more secure than professional software because vulnerabilities have been discovered and corrected by a large community. Professional developers have only a small number of people working on problems and have mixed motives with respect to informing the public about security concerns.

It is good practice to avoid deception in any Web-based research, at all costs. One of the ethical reasons to do this is to keep a good reputation for Internet-based research. An ethics review board is concerned for the ethical treatment of participants, and certain deceptions would not be considered harmful to participants. However, studies involving deception might be harmful to other researchers who do not practice deception if Internet research should acquire a bad reputation. Therefore, one should consider the impact of one's research on other scientists (and indirectly on society) before one does anything to give Internet-based research a bad name. For more information on issues of deception and ethics of Web research, see Birnbaum (2004b) and Barchard and Williams (2008).

In addition to ethical concerns in recruiting, there are issues of good manners and good taste. It is considered impolite to send people attachments they did not request. People now fear to open them because they may carry a commercial message, a worm, or a computer virus. And large attachments can fill up mailboxes, can be slow to download, and are generally seen as a kind of injury (or threat of injury) that you have inflicted on your recipients. Keep in mind that computers break down all the time. Suppose you opened an attachment from someone, and then your computer broke down. It would be natural to assign blame and seek justice, even though the event was mere coincidence. By means of the Internet, it is easy to reach a large number of people for good or ill, and therefore it is possible to annoy if not anger large numbers of people.

Instead of sending attachments, send a link to the file that you have uploaded to the Web.

In your messages recruiting participants or reporting availability of results, you should include your correct name, e-mail and postal address, telephone number, and affiliation. Behaving politely on the Internet is not only a nice thing to do, it is also a necessity for the Internet researchers to avoid being confused with purveyors of SPAM, spoofers (senders of e-mail from fake addresses), or even worse. The impolite and ignorant behaviors of a few spammers and malicious hackers have already destroyed much of the good spirit and free flow of communication on the Internet. E-mail is now filtered and even innocent messages and innocent attachments are disappearing into e-mail void, as institutions try to protect themselves from the massive number of unwanted or unsolicited e-mails.

27.9.3.2 Polite Recruitment

Similarly, it is important to be very careful when recruiting participants for your Internet-based study. Let others check the wording of your invitation, before you send it off or post it. If you recruit participants via a mailing list or newsgroup, then first have the moderator of the list approve your posting. An even better solution is to ask someone else, such as the manager of a group, to send the announcement for you. That way, people will know that the announcement is legitimate, and if they consider your invitation to be spam, they will blame the moderator's judgment rather than yours.

Do not do anything that resembles a chain letter, commercial advertising, or spamming. Internet researchers have been "blacklisted" and "e-mail bombed" for recruiting too aggressively. Such cases are not only embarrassing and time-consuming (responding to "flames" or angry messages attacking one's manners), they also pose a threat to the reputation of the entire community. If you are in doubt that people want to receive your message, it is probably best not to send it. Consider recruiting participants in the least intrusive manner, which is probably done by posting the announcement at designated Web sites mentioned earlier in this chapter. These sites are explicitly designated to potential participants to find studies and to researchers to find participants.

A set of considerations ("rules") for proper behavior on the Internet can be found at http://www.albion.com/netiquette/ and Internet research specific advice is available from Michalak and Szabo (1998).

27.10 WHERE TO LEARN MORE

In addition to journals and professional books on the topic, there are two organizations holding conferences that provide good opportunities to meet and interact with people involved in psychological Internet-based research. These are the General Online Research Conference (GOR), established in 1997 and hosted jointly by the German Society for Online Research (DGOF) and local organization teams that differ from year to year. More than 350 participants attended each GOR conference in the past few years.

The central purpose of GOR conferences is to provide a venue for the presentation of empirical studies that examine the Internet as both an instrument for, and an object of, scientific investigation. Topics of investigation presented include (but are not limited to): quality of Internet-based research, Internet research methods in market research and polling, computer-mediated communication, Web experiments, and marketing on the Internet.

Society for Computers in Psychology (SCiP) meetings take place on an annual basis, in mid-November and are scheduled to occur in the same location as the Psychonomic Society and Society for Judgment and Decision Making (JDM) meetings, on the day preceding the annual meeting of the Psychonomic Society. This organization is somewhat older than GOR, and is devoted not only to Internet-based research but also to other applications of computers in psychological research and teaching. The primary journal of this organization is *Behavior Research Methods*. Another journal we recommend is the *International Journal of Internet Science*, a free open access journal available from http://www.ijis.net.

REFERENCES

American Psychological Association. 2010. *Publication Manual of the American Psychological Association*, 6th ed. Washington, DC: Author.

Barchard, K. A., and J. E. Williams. 2008. Practical advice for conducting ethical online experiments and questionnaires for United States psychologists. *Behavior Research Methods* 40: 1111–1128.

Baron, J., and M. Siepmann. 2000. Techniques for creating and using web questionnaires in research and teaching. In *Psychological experiments on the Internet*, ed. M. H. Birnbaum, 235–265. San Diego, CA: Academic Press.

Berker, T. 1999. Online-offline: Methodologische Erträge einer Internetnutzungsstudie zur Klärung einer Zentralkategorie des Online-Research [On-line versus off-line: Methodological earnings of a study on Internet use devised to clarify a central category of on-line research]. In *Current Internet Science—Trends, Techniques, Results. Aktuelle Online Forschung—Trends, Techniken, Ergebnisse*, eds. U.-D. Reips et al. Zürich: Online Press, available at http://dgof.de/tband99/.

Berker, T. 2002. World Wide Web use at a German university—computers, sex, and imported names: results of a log file analysis. In *Online Social Sciences*, eds. B. Batinic, U.-D. Reips, and M. Bosnjak, 365–382. Göttingen, Germany: Hogrefe.

Birnbaum, M. H. 1999a. How to show that 9 > 221: Collect judgments in a between-subjects design. *Psychological Methods* 4(3): 243–249.

Birnbaum, M. H. 1999b. Testing critical properties of decision making on the Internet. *Psychological Science* 10: 399–407.

Birnbaum, M. H. ed. 2000a. *Psychological Experiments on the Internet*. San Diego: 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(2): 339–346.

Birnbaum, M. H. 2001a. *Introduction to Behavioral Research on the Internet*. Upper Saddle River, NJ: Prentice-Hall.

Birnbaum, M. H. 2001b. A Web-based program of research on decision making. In *Dimensions of Internet Science*, eds. U.-D. Reips and M. Bosnjak, 23–55. Lengerich, Germany: Pabst Science.

Birnbaum, M. H. 2004a. Human research and data collection via the Internet. *Annual Review of Psychology* 55: 803–832.

Birnbaum, M. H. 2004b. Methodological and ethical issues in conducting social psychology research via the Internet. In *Handbook of Methods in Social Psychology*, eds. C. Sansone, C. C. Morf, and A. T. Panter, 359–382. Thousand Oaks, CA: Sage.

Birnbaum, M. H. 2007. Designing online experiments. In *The Oxford Handbook of Internet Psychology*, eds. A. Joinson et al., 391–403. Oxford, UK: Oxford University Press.

Birnbaum, M. H., and B. A. Mellers. 1989. Mediated models for the analysis of confounded variables and self-selected samples. *Journal of Educational Statistics* 14: 146–158.

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

Buchanan, T. 2000. Potential of the Internet for personality research. In *Psychological Experiments on the Internet*, ed. M. H. Birnbaum, 121–140. San Diego, CA: Academic Press.

Buchanan, T. 2001. Online personality assessment. In *Dimensions of Internet Science*, eds. U.-D. Reips and M. Bosnjak, 57–74. Lengerich, Germany: Pabst Science.

Dillman, D. A., and D. K. Bowker. 2001. The Web questionnaire challenge to survey methodologists. In *Dimensions of Internet Science*, eds. U.-D. Reips and M. Bosnjak, 159–178. Lengerich, Germany: Pabst Science.

Dillman, D. A., J. D. Smyth, and L. M. Christian. 2008. *Internet, Mail, and Mixed-mode Surveys: The Tailored Design Method*. New York: Wiley.

Drake, P. J. W. 2001. Successful aging: investment in genealogy as a function of generativity, mobility and sense of place. Master of Arts Thesis. California State University, Fullerton.

Eichstaedt, J. 2001. An inaccurate-timing filter for reaction-time measurement by JAVA-applets implementing Internet-based experiments. *Behavior Research Methods, Instruments, and Computers* 33: 179–186.

Frick, A., M. T. Bächtiger, and U.-D. Reips. 2001. Financial incentives, personal information, and drop-out in online studies. In *Dimensions of Internet Science*, eds. U.-D. Reips and M. Bosnjak, 209–219. Lengerich: Pabst Science.

Fraley, R. C. 2004. *How to Conduct Behavioral Research over the Internet: A Beginner's Guide to HTML and CGI/Perl*. New York: The Guilford Press.

Francis, G., I. Neath, and A. M. Surprenant. 2000. The cognitive psychology online laboratory. In *Psychological Experiments on the Internet*, ed. M. H. Birnbaum, 267–283. San Diego, CA: Academic Press.

Funke, F., and U.-D. Reips. 2007. Messinstrumente und Skalen [Measuring devices and scales]. In *Online-Forschung 2007: Grundlagen und Fallstudien*, eds. M. Welker and O. Wenzel, 52–76. Köln: Herbert von Halem.

Garrett, J. J. 2005. Ajax: A new approach to Web applications. http://www.adaptivepath.com/ideas/essays/archives/000385.php (accessed Feb. 8, 2010).

Göritz, A. S. 2006. Web-based data collection. Retrieved April 10, 2010, from http://www.goeritz.net/ati/material.pdf.

Göritz, A. S. 2007. Using online panels in psychological research. In *The Oxford Handbook of Internet Psychology*, eds. A. N. Joinson et al., 473–485. Oxford, UK: Oxford University Press.

Göritz, A. S. 2009. Building and managing an online panel with PHP PanelAdmin. *Behavior Research Methods* 41(4): 1177–1182.

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

Hardin, C., and M. H. Birnbaum. 1990. Malleability of "ratio" judgments of occupational prestige. *American Journal of Psychology* 103: 1–20.

Knapp, F., and M. Heidingsfelder. 2001. Drop-out analysis: effects of the survey design. In *Dimensions of Internet Science*, eds. U.-D. Reips and M. Bosnjak, 221–230. Lengerich, Germany: Pabst Science.

Krantz, J. H. 2001. Stimulus delivery on the Web: What can be presented when calibration isn't possible? In *Dimensions of Internet Science*, eds. U.-D. Reips and M. Bosnjak, 113–130. Lengerich, Germany: Pabst Science.

Krantz, J. H., J. Ballard, and J. Scher. 1997. Comparing the results of laboratory and World-Wide Web samples on the determinants of female attractiveness. *Behavior Research Methods, Instruments, & Computers*, 29: 264–269.

Krantz, J. H., and R. Dalal. 2000. Validity of Web-based psychological research. In *Psychological Experiments on the Internet*, eds. M. H. Birnbaum, 35–60. San Diego, CA: Academic Press.

Krantz, J. H., and J. E. Williams. 2010. Using graphics, photographs, and dynamic media. In *Advanced Methods for Behavioral Research on the Internet*, eds. S. Gosling and J. A. Johnson, 45–61. Washington, DC: American Psychological Association.

Mangan, M., and Reips, U.-D. 2007. Sleep, sex, and the Web: Surveying the difficult-to-reach clinical population suffering from sexsomnia. *Behavior Research Methods, 39*, 233–236.

McGraw, K. O., M. D. Tew, and J. E. Williams. 2000. PsychExps: An on-line psychology laboratory. In *Psychological experiments on the Internet*, ed. M. H. Birnbaum, 219–233. San Diego, CA: Academic Press.

Michalak, E. E., and A. Szabo. 1998. Guidelines for Internet research: an update. *European Psychologist* 3(1): 70–75.

Musch, J., A. Bröder, and K. C. Klauer. 2001. Improving survey research on the World-Wide Web using the randomized response technique. In *Dimensions of Internet Science*, eds. U.-D. Reips and M. Bosnjak, 179–192. Lengerich: Pabst Science Publishers.

Musch, J., and U.-D. Reips. 2000. A brief history of Web experimenting. In *Psychological Experiments on the Internet*, ed. M. H. Birnbaum, 61–87. San Diego, CA: Academic Press.

Parducci, A. 1995. *Happiness, Pleasure, and Judgment*. Mahwah, NJ: Lawrence Erlbaum.

Reips, U.-D. 1995. The Web experiment method. http://www.uni-tuebingen.de/uni/sii/Ulf/Lab/wwwExpMethod.html, now at http://wexlab.eu/wwwExpMethod.html (accessed Sept. 1, 1995).

Reips, U.-D. 1997. Das psychologische Experimentieren im Internet. In *Internet für Psychologen*, ed. B. Batinic, 245–265. Göttingen, Germany: Hogrefe.

Reips, U.-D. 1999. Online Research with children. In *Current Internet Science—Trends*, *Techniques, Results*, eds. U.-D. Reips et al., http://gor.de/gor99/tband99/pdfs/q_z/reips.pdf. Zürich: Online Press.

Reips, U.-D. 2000. The Web experiment method: Advantages, disadvantages, and solutions. In *Psychological experiments on the Internet*, ed. M. H. Birnbaum, 89–117. San Diego, CA: Academic Press.

Reips, U.-D. 2001a. Merging field and institution: running a Web laboratory. In *Dimensions of Internet Science*, eds. U.-D. Reips and M. Bosnjak, 1–22. Lengerich, Germany: Pabst Science Publishers.

Reips, U.-D. 2001b. The Web Experimental Psychology Lab: Five years of data collection on the Internet. *Behavior Research Methods, Instruments, & Computers* 33: 201–211.

Reips, U.-D. 2002a. Context effects in Web surveys. In *Online Social Sciences*, eds. B. Batinic, U.-D. Reips, and M. Bosnjak, 69–79. Göttingen, Germany: Hogrefe & Huber.

Reips, U.-D. 2002b. Internet-based psychological experimenting: Five do's and five don'ts. *Social Science Computer Review* 20(3): 241–249.

Reips, U.-D. 2002c. Standards for Internet-based experimenting. *Experimental Psychology* 49(4): 243–256.

Reips, U.-D. 2002d. Theory and techniques of conducting Web experiments. In *Online Social Sciences*, eds. B. Batinic, U.-D. Reips, and M. Bosnjak, 219–249. Seattle: Hogrefe & Huber.

Reips, U.-D. 2006. Web-based methods. In *Handbook of Multimethod Measurement in Psychology*, eds. M. Eid and E. Diener, 73–85. Washington, DC: American Psychological Association.

Reips, U.-D. 2007. The methodology of Internet-based experiments. In *The Oxford Handbook of Internet Psychology*, eds. A. Joinson et al., 373–390. Oxford: Oxford University Press.

Reips, U.-D. 2010. Design and formatting in Internet-based research. In *Advanced Internet Methods in the Behavioral Sciences*, eds. S. Gosling and J. A. Johnson, 29–43. Washington, DC: American Psychological Association.

Reips, U.-D. in press. Using the internet to collect data. In *APA Handbook of Research Methods in Psychology*, eds H. Cooper, P. Camic, R. Gonzalez, D. Long, and A. Panter. Washington, DC: American Psychological Association.

Reips, U.-D., and M. Bosnjak, eds. 2001. *Dimensions of Internet Science*. Lengerich, Germany: Pabst Science.

Reips, U.-D., T. Buchanan, A. Joinson, and C. Paine. 2010. Internet questionnaires in e-health contexts: Non-response to sensitive items. Manuscript submitted for publication.

Reips, U.-D., and L. Franek. 2004. Mitarbeiterbefragungen per Internet oder Papier? Der Einfluss von Anonymität, Freiwilligkeit und Alter auf das Antwortverhalten [Employee surveys via Internet or paper? The influence of anonymity, voluntariness and age on answering behavior]. *Wirtschaftspsychologie* 6(1): 67–83.

Reips, U.-D., and F. Funke. 2008. Interval level measurement with visual analogue scales in Internet-based research: VAS Generator. *Behavior Research Methods* 40: 699–704.

Reips, U.-D., and J. Krantz. 2010. Conducting true experiments on the Web. In *Advanced Internet Methods in the Behavioral Sciences*, eds. S. Gosling and J. Johnson, (193–216). Washington, DC: American Psychological Association.

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

Rose, B. J., and M. H. Birnbaum. 1975. Judgments of differences and ratios of numerals. *Perception & Psychophysics* 18: 194–200.

Schmidt, W. C. 1997a. World-Wide Web survey research: Benefits, potential problems, and solutions. *Behavioral Research Methods, Instruments, & Computers* 29: 274–279.

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

Schmidt, W. C. 2000. The server-side of psychology Web experiments. In *Psychological Experiments on the Internet*, ed. M. H. Birnbaum, 285–310. San Diego, CA: Academic Press.

Schmidt, W. C. 2007. Technical considerations when implementing online research. In *The Oxford Handbook of Internet Psychology*, eds. A. Joinsonet al., 461–472. Oxford: Oxford University Press.

Schmidt, W. C., R. Hoffman, and J. MacDonald. 1997. Operate your own World-Wide Web server. *Behavior Research Methods, Instruments, & Computers* 29: 189–193.

Schwarz, S., and U.-D. Reips. 2001. CGI versus JavaScript: A Web experiment on the reversed hindsight bias. In *Dimensions of Internet Science*, eds. U.–D. Reips and M. Bosnjak, 75–90. Lengerich: Pabst.

Smith, M. A., and B. Leigh. 1997. Virtual subjects: Using the Internet as an alternative source of subjects and research environment. *Behavior Research Methods, Instruments, & Computers* 29: 496–505.

Smyth, J. D., D. A. Dillman, L. M. Christian, and M. J. Stern. 2006. Effects of using visual design principles to group response options in Web surveys. *International Journal of Internet Science* 1: 6–16.

Stevenson, M. K. 1993. Decision making with long-term consequences: Temporal discounting for single and multiple outcomes in the future. *Journal of Experimental Psychology: General* 122: 3–22.

Stieger, S., and U.-D. Reips. 2010. What are participants doing while filling in an online questionnaire: A paradata collection tool and an empirical study. *Computers in Human Behavior* 26: 1488–1495.

Tuten, T. L., M. Bosnjak, and W. Bandilla. 2000. Banner-advertised Web surveys. *Marketing Research* 11(4): 17–21.

Tuten, T. L., D. J. Urban, and M. Bosnjak. 2002. Internet surveys and data quality: A review. In *Online Social Sciences,* eds. B. Batinic, U.-D. Reips, and M. Bosnjak, 7–26. Seattle, WA: Hofrefe & Huber.

Welch, N., and J. H. Krantz. 1996. The world-wide Web as a medium for psychoacoustical demonstrations and experiments: Experience and results. *Behavior Research Methods, Instruments, & Computers* 28: 192–196.

Wikibooks, n.d. Mac OS X Tiger/Using your Mac as a Web Server. http://en.wikibooks.org/wiki/Mac_OS_X_Tiger/Using_your_Mac_as_a_Web_Server (accessed March 24, 2010).

Wikipedia. 2010. Ajax (programming). http://en.wikipedia.org/w/index.php?title=Ajax_(programming)&oldid=352212625 (accessed Jan. 27, 2010).

Williams, J. E., K. O. McGraw, and M. D. Tew. 1999. Undergraduate labs and computers: The case for PsychExps. *Behavioral Research Methods, Instruments, and Computers* 31(2): 287–291.