Programming online experiments with jsPsych

Joshua R. de Leeuw (jodeleeu@indiana.edu)
Department of Psychological and Brain Science & Program in Cognitive Science
Indiana University, 1101 E. 10th St.
Bloomington, IN 47405 USA

Keywords: online experiments; methodology; jsPsych; JavaScript

Overview

This tutorial is an introduction to jsPsych, which is a free and open-source software package for creating experiments that run in a web browser. Participants in the tutorial will learn how to build an experiment using jsPsych, and how to extend and customize jsPsych for novel experimental paradigms.

Running experiments online is a popular method among cognitive scientists. Data collection is (extremely) fast and cheap, and the quality of the data is generally quite high (Buhrmester, Kwang, & Gosling, 2011; Crump, McDonnell, & Gureckis, 2013; Simcox & Fiez, 2014; Zwaan & Pecher, 2012). There are methodological benefits as well, such as a more diverse subject pool (Arnett, 2008; Ross, Irani, Silberman, Zaldivar, & Tomlinson, 2010) and a reduction in possible experimenter-induced biases.

Building an experiment that can be run online requires proficiency in web development techniques that many researchers lack. As a result, there is a demand for tools that make online experiments easier to develop and run. A few such tools are now available and used within the cognitive science community, including PsiTurk (McDonnell et al., 2012), QRTEngine (Barnhoorn, Haasnoot, Bocanegra, & van Steenbergen, 2014), and jsPsych (de Leeuw, 2014).

The main benefit of using jsPsych is that it reduces the complexity of programming experiments for the web. Researchers using jsPsych will still need to know how to program (in JavaScript), but the programming tasks will map more naturally on to the design (rather than the implementation) of the experiment. For example, it's not necessary to write code that will determine what key was pressed and what the response time is. jsPsych handles this, and other functionality that is common across most experiments, such as figuring out which task/trial to run next, controlling the flow of the participant through the study, storing data, and so on. However, it is necessary to describe, in code, the design of the experiment, including what kinds of tasks the subject will complete, what stimuli they will see, how long displays will last, and so on.

Experiments in jsPsych are composed of individual tasks, such as showing the subject instructions, displaying a stimulus and getting a response, or filling out a survey question. These tasks are assembled, by the researcher, into a timeline. A timeline describes the tasks, the parameters for the tasks, and what order the tasks will occur. A main design feature of jsPsych is that each task is defined in its own code file, known as a plugin. Plugins have a standardized, yet extremely flexible, structure. This makes it possible to create custom plugins for tasks that are not possible with the set of plugins included in jsPsych. It is also easy to share plugins, to make replications and further manipulations of a particular task relatively easy to implement for other researchers.

Information about jsPsych was presented at the 2014 Cognitive Science Society meeting as part of a larger tutorial about creating online experiments (de Leeuw et al., 2014). This tutorial will go into significantly more depth, covering more features of the library, how to develop new tasks/plugins, and demonstrating a set of new features that were added in a major update in October 2014. This update made it possible to implement a variety of different experimental designs that were previously impossible with jsPsych, including conditional branching and looping structures. Other new features from the update include the ability to easily randomize trial order, repeat sets of trials, and automatically display a progress bar. For a complete list of features, see the online documentation at http://docs.jspsych.org.

The tutorial is targeted at researchers who have some familiarity and comfort with programming and an interest in developing experiments for the web. Researchers who have no programming background may find it difficult to follow along, as the basics of programming won't be covered. The tutorial should be of interest to researchers with all levels of web-development expertise. Those who are less familiar with web-development techniques will find it easier to learn to use jsPsych than learning to create experiments from scratch, while researchers with a web-development background may find that jsPsych offers a streamlined way of building experiments that is more efficient than programming experiments on a case-by-case basis. Participants are strongly encouraged to bring a laptop with a programming-friendly text editor, such as Atom (http://www.atom.io), to follow along, but may also find it informative to just observe and learn about what is possible with jsPsych.

Summary of Tutorial Content

The morning session of the tutorial will be focused on describing the capabilities of jsPsych and how to use the software. In the afternoon, tutorial participants will have the opportunity to work hands-on building a jsPsych experiment. Participants are encouraged to work on developing experiments for their own research, and should

bring materials such as stimuli needed to assemble the experiment.

The morning presentation will be divided in three parts. The first part will be a lecture-style tour of jsPsych. This will include describing the conceptual design goals behind jsPsych, what problems it aims to solve, and situations in which it is and isn't a useful tool. jsPsych will be compared to other tools that are available for online research, to highlight the relative strengths and weaknesses of each. A variety of jsPsych experiments will be demonstrated, to give participants an idea of what's possible with the library. Data from a recent experiment validating the response time accuracy of jsPsych, and JavaScript response time measurement in general, will also be covered (de Leeuw & Motz, in press).

The second part will be a hands-on activity in which participants assemble a jsPsych experiment from start to finish. This part of the tutorial will cover the basics of working with jsPsych all the way through advanced features of the library such as conditional looping structures. By the end of this part, participants will have had the opportunity to build a simple experiment that uses a number of different features of the library.

The final part will discuss how to extend and customize jsPsych. The main focus of this part will be explaining how to create a new jsPsych plugin, which enables researchers to program virtually any computer-based task and include it within the framework of jsPsych. Customizing jsPsych allows researchers to take advantage of the numerous features of jsPsych, while still programming their own tasks that may not be possible with the set of tasks that jsPsych includes.

Materials from the tutorial, including code files and presentation slides, will be made available online.

Presenter

Josh de Leeuw is a graduate student at Indiana University. He is the creator of jsPsych. In addition to using jsPsych for nearly all of his own research, he regularly provides assistance and advice to researchers who are using the platform. He has presented several talks and tutorials about jsPsych, including at the 2014 Cognitive Science Society meeting in Quebec City (de Leeuw et al., 2014).

References

Arnett, J. J. (2008). The neglected 95%: Why American psychology needs to become less American. *The American Psychologist*, *63*(7), 602–14. doi:10.1037/0003-066X.63.7.602

Barnhoorn, J. S., Haasnoot, E., Bocanegra, B. R., & van Steenbergen, H. (2014). QRTEngine: An easy solution for running online reaction time experiments using Qualtrics. *Behavior Research Methods*, Advance Online Publication.

Buhrmester, M., Kwang, T., & Gosling, S. D. (2011). Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, *6*(1), 3–5. doi:10.1177/1745691610393980

Crump, M. J. C., McDonnell, J. V, & Gureckis, T. M. (2013). Evaluating Amazon's Mechanical Turk as a tool for experimental behavioral research. *PloS ONE*, *8*(3), e51382. doi:10.1371/journal.pone.0057410

de Leeuw, J. R. (2014). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior Research Methods*, Advance Online Publication. doi:10.3758/s13428-014-0458-y

de Leeuw, J. R., Coenen, A., Markant, D., Martin, J. B., McDonnell, J. V, Rich, A. S., & Gureckis, T. M. (2014). Online Experiments using jsPsych, psiTurk, and Amazon Mechanical Turk. In P. Bello, M. Guarini, M. McShane, & B. Scassellati (Eds.), *Proceedings of the 36th Annual Conference of the Cognitive Science Society* (pp. 41–42). Austin, TX: Cognitive Science Society.

de Leeuw, J. R., & Motz, B. A. (in press). Psychophysics in a web browser? Comparing response times collected with JavaScript and Psychophysics Toolbox in a visual search task. *Behavior Research Methods*.

McDonnell, J., Martin, J. B., Markant, D. B., Coenen, A., Rich, A. S., & Gureckis, T. M. (2012). psiTurk. New York, NY: New York University. Retrieved from https://github.com/NYUCCL/psiTurk

Ross, J., Irani, L., Silberman, M. S., Zaldivar, A., & Tomlinson, B. (2010). Who are the crowdworkers? Shifting demographics in Mechanical Turk. In *CHI EA '10* (pp. 2863–2872). New York, NY: ACM.

Simcox, T., & Fiez, J. A. (2014). Collecting response times using Amazon Mechanical Turk and Adobe Flash. *Behavior Research Methods*, *46*(1), 95–111. doi:10.3758/s13428-013-0345-y

Zwaan, R. A., & Pecher, D. (2012). Revisiting mental simulation in language comprehension: six replication attempts. *PloS ONE*, *7*(12), e51382. doi:10.1371/journal.pone.0051382