

Online Experiments using jsPsych, psiTurk, and Amazon Mechanical Turk

Joshua R. de Leeuw (jodeleew@indiana.edu)

Department of Psychological and Brain Sciences, Program in Cognitive Science, Indiana University,
Bloomington, IN 47405

Anna Coenen (ac4066@nyu.edu)

Doug Markant (doug.markant@nyu.edu)

Jay B. Martin (jbmartin@nyu.edu)

John V. McDonnell (jmcdon10@gmail.com)

Alexander S. Rich (asr443@nyu.edu)

Todd M. Gureckis (todd.gureckis@nyu.edu)

Department of Psychology, New York University
New York, NY 10003

Keywords: Amazon Mechanical Turk; Online Experiments;
jsPsych; psiTurk; Open Science

Objectives

This half-day tutorial will cover how to build and deploy online experiments using jsPsych, psiTurk, and Amazon Mechanical Turk (AMT). jsPsych is an open-source JavaScript library that facilitates building behavioral experiments in a web browser. psiTurk is an open-source Python platform that simplifies the process of running an experiment using AMT. Together, these two software packages reduce the complexity of setting up an online experiment on AMT, enabling researchers with little software programming experience to take advantage of online experiments. By the end of the tutorial, participants will have gained hands-on experience in programming and deploying a basic behavioral experiment on AMT.

Researchers in the Cognitive Science community have been using AMT, and online experiments in general, for several years, but the learning curve can be steep for researchers who are not familiar with web development. While some tools exist for certain kinds of simple experiments (e.g. questionnaires), programming more complex experiments with dynamic elements requires knowledge of web-oriented programming. The tools covered in this tutorial simplify the process of programming online experiments, opening up the possibility of conducting online experiments to more researchers in the cognitive science community.

Workshops covering AMT (Mason & Suri, 2011) and psiTurk (Coenen, Markant, Martin, & McDonnell, 2013) have been offered at previous Cognitive Science Society meetings. This tutorial goes one step further by covering jsPsych as well. Together, these tools cover the entire process of assembling and running an online experiment.

Outline of the Tutorial

Participants at the tutorial will be invited to work hands-on with the creation of a simple online experiment that demonstrates the principles behind jsPsych and psiTurk. The tutorial will be organized in four parts.

Part 1: What does the research say about online data collection?

This section of the tutorial will be a brief introduction to some of the issues surrounding AMT and online experiments, but the focus of the tutorial will be parts 2 and 3.

Online experiments are appealing for a number of reasons: faster data collection, lower costs, access to a different subject pool, and improved anonymity of subjects and experimenters are some of the most commonly named. However, online experiments give up some of the control of a laboratory environment, leading to concerns about the quality of the data. There are now several published results that compare AMT experiments to their laboratory counterparts (Paolacci, Chandler, & Ipeirotis, 2010; Buhrmester, Kwang, & Gosling, 2011; Zwann & Pechler, 2012; Crump, McDonnell, & Gureckis, 2013; Goodman, Cryder, & Cheema, 2013). In the tutorial, we will summarize these findings and their implications for running AMT based experiments.

Part 2: Assembling an experiment with jsPsych

jsPsych (<http://www.jspsych.org>) is an open-source JavaScript library that simplifies the process of writing a web-browser-based experiment (de Leeuw, 2014). jsPsych contains a core library, which serves as the engine to run experiments, and a set of plugins, each of which defines a different kind of trial that a subject in an experiment might do. For example, there are plugins for displaying instructions, showing stimuli and collecting responses via the keyboard, and displaying a consent form. Assembling an experiment with jsPsych involves putting together the different plugins that are needed and specifying the parameters of those plugins (such as what stimuli to show and how long to show them). These plugins can be assembled to create many different behavioral tasks that are of interest to cognitive scientists.

jsPsych can also be extended by writing new plugins. The structure of a plugin is flexible enough to permit most kinds of computer-based tasks. Because plugins are individual, stand-alone components of the library, each plugin can be combined with any of the others. As researchers use and extend jsPsych

to cover new kinds of tasks, the library grows making it easier for other researchers to create new experiments.

This part of the tutorial will cover how to build an experiment using existing jsPsych plugins. We will show how the same plugin can be reused for different kinds of experiments, to highlight the flexibility of the library. We will give a short overview of how to write new plugins for jsPsych, but this will not be the main focus of the tutorial.

Part 3: Launching an experiment on AMT with psiTurk

psiTurk (<http://psiturk.org>) is an open source framework for running experiments on Amazon Mechanical Turk. One of the major hurdles to running online experiments is the need for the installation and administration of web server (e.g., Apache). psiTurk obviates this requirement by allowing people to run experiments on any computer they choose including their personal laptop or office computer. In addition, psiTurk simplifies many aspects of running experiments on Mechanical Turk. For example, psiTurk automatically assigns participants to conditions in a random fashion, it allows researchers to quickly pay and assign bonuses to workers with a few simple commands, and tracks when users do other activities within their browser or computer and provides this data in an easy to analyze database. These and other features will be covered in detail in our tutorial. Finally, we will show how jsPsych and psiTurk can be combined to take full advantage of both software packages.

Part 4: Advancing open science with psiTurk's Experiment Exchange

One of the major goals of psiTurk is to enable frictionless replicability of research studies. Currently, researchers using AMT to conduct experiments all use a different set of script or tools. As a result, sharing the code for an experiment with another researcher can be fairly complex and require lots of documentation. psiTurk aims to provide a powerful and general set of tools which will encourage researchers to adopt a "standard" way of writing online experiments. This will increase the replicability of findings since it will be easier to download and run another researcher's experiment. psiTurk facilitates this via an "Experiment Exchange" website (<http://psiturk.org/ee>) which enables researchers to share their psiTurk compatible experiments with one another. Ultimately, it is possible with psiTurk to download another person's experiment and start collecting online data in that experiment within a few minutes.

Audience

This tutorial is aimed at researchers who are interested in conducting online experiments and would like to gain experience with tools that will simplify the complexity of developing and running such experiments. As the tools being covered are quite flexible, they will be useful to researchers who are interested in collecting behavioral data in a wide range of areas.

The hands-on portion of the tutorial will be aimed at people who have some basic familiarity with web programming, though we will provide materials to help novices get up and running.

Participants who wish to work hands-on during the tutorial should bring a laptop. jsPsych will work with any up-to-date version of the major web browsers (Chrome, Firefox, Safari, Internet Explorer), and only requires a basic text editor that can save HTML files (e.g. Notepad++ for Windows, TextMate for OSX). psiTurk works best on Unix-based platforms (OSX, Linux), but we will show non-Unix users how to use free tools online to run psiTurk on a remote system.

Presenters

The presenters have all used online experiments and AMT extensively in their research. All of them are authors of either psiTurk or jsPsych. As there are several presenters, there will be opportunities for presenters who are not actively presenting to provide one-on-one assistance as needed.

References

- Buhrmester, M., Kwang, T., & Gosling, S. (2011). Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1), 3–5.
- Coenen, A., Markant, D., Martin, J., & McDonnell, J. (2013). Using mechanical turk and psiturk for dynamic web experiments. In M. Knauff, M. Pauen, N. Sebanz, & I. Wachsmuth (Eds.), *Proceedings of the 35th annual conference of the cognitive science society* (pp. 22–23). Austin, TX: Cognitive Science Society.
- Crump, M., McDonnell, J., & Gureckis, T. (2013). Evaluating Amazon's Mechanical Turk as a tool for experimental behavioral research. *PLoS ONE*, 8(3).
- de Leeuw, J. (2014). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior Research Methods*, Advance online publication.
- Goodman, J., Cryder, C., & Cheema, A. (2013). Data collection in a flat world: The strengths and weaknesses of Mechanical Turk samples. *Journal of Behavioral Decision Making*, 26(1), 213–224.
- Mason, W., & Suri, S. (2011). How to use mechanical turk for cognitive science research. In L. Carlson, C. Hölscher, & T. Shipley (Eds.), *Proceedings of the 33rd annual conference of the cognitive science society* (pp. 66–67). Austin, TX: Cognitive Science Society.
- Paolacci, G., Chandler, J., & Ipeirotis, P. (2010). Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5), 411–419.
- Zwann, R., & Pechler, D. (2012). Revisiting mental simulation in language comprehension: six replication attempts. *PLoS ONE*, 7(3).