METHODS SHOWCASE ARTICLE

Conducting Linguistic Experiments Online With OpenSesame and OSWeb

Sebastiaan Mathôt and Jennifer March

Abstract: In this Methods Showcase Article, we outline a workflow for running behavioral experiments online, with a focus on experiments that rely on presentation of complex stimuli and measurement of reaction times, which includes many psycholinguistic experiments. The workflow that we describe here relies on three tools: OpenSesame/OSWeb (open source) provides a user-friendly graphical interface for developing experiments; JATOS (open source) is server software for hosting experiments; and Prolific (commercial) is a platform for recruiting participants. These three tools integrate well with each other and together provide a workflow that requires little technical expertise. We discuss, and illustrate through an example study, several challenges that are associated with running online experiments, including temporal precision, the implementation of counterbalancing, data quality, and issues related to privacy and ethics. We conclude that these challenges are real but surmountable, and that in many cases online experiments are a viable alternative to laboratory-based experiments.

Keywords experimental design; data collection; online experiments; methods showcase

Introduction

Traditionally, participants in behavioral psycholinguistic experiments are asked to visit a psychological laboratory. There, a researcher welcomes them and provides them with a consent form. After having signed this form,
participants enter a small cubicle where an experiment, typically a computer task, is waiting for them. After having completed the experiment, participants leave the cubicle and inform the researcher, who subsequently enters the cubicle to retrieve the data.

The software landscape for conducting such lab-based experiments has slowly shifted over the years. In the early 2000s, most labs were using proprietary software such as E-Prime, Inquisit, SuperLab, DirectRT, and Presentation (Stahl, 2006). With time, the software landscape has become more varied, with many labs moving toward open-source software, with PsychtoolBox (Brainard, 1997) being the first on the scene, followed later by PsychoPy (Peirce, 2007) and OpenSesame (Mathôt, Schreij, & Theeuwes, 2012). However, despite the shift toward new tools and software, the lab-based approach remained the default for many years, and to a considerable extent it still is. Of the experiments reported in Language Learning during 2020 and 2021, only six experiments were conducted online, compared to 34 experiments that were conducted in laboratory or laboratory-like settings (see https://osf.io/ywbej/ for an overview).

In a typical online experiment, participants use a web browser to visit a web page that contains the experiment. The main appeal of running experiments online is the potential for collecting large amounts of data in short periods of time. With online experiments, researchers no longer need to welcome participants into the laboratory one at a time; researchers only need to post a link on a participant recruitment platform, such as Prolific (https://www.prolific.co). Consider a psycholinguistic “megastudy,” such as MEGALEX (Ferrand et al., 2018), in which 197 participants spent 20 hr each (in 20-min sessions distributed over multiple weeks following a flexible schedule) performing a lexical-decision task in laboratory cubicles across various institutes. Running such a project online would have (certainly) reduced the expenses associated with hiring research assistants, travel time for participants, and maintaining laboratory space, with (hopefully) little impact on data quality. This is the promise of online experiments.

When it comes to online experiments, it is important to make a distinction between questionnaire-based experiments and experiments that rely on presentation of complex stimuli and measurement of reaction times. Questionnaire-based experiments, and more generally experiments that can be implemented with regular web forms, have already been conducted online for some years (Munro et al., 2010). These include many linguistic experiments in which participants make judgments about words or sentences through rating scales or multiple-choice selections. Such experiments can be implemented
with user-friendly online questionnaire software such as Qualtrics (https://www.qualtrics.com), SurveyMonkey (https://www.surveymonkey.com), or even Google Forms (https://docs.google.com/forms).

However, the situation is different for the kinds of experiments that will be the focus of this Methods Showcase Article: experiments that rely on presentation of complex stimuli and measurement of reaction times, such as many psycholinguistic experiments (reviewed in Kaiser, 2013). These types of experiments have only recently begun to be conducted online on a large scale. So far, three main factors have been holding back widespread adoption of online experiments of this kind. The first limiting factor is that, until recently and in contrast to questionnaire-based experiments, developing such experiments required the use of advanced web technologies such as JavaScript (see, e.g., an early series of online studies by Crump, McDonnell, & Gureckis, 2013), which is the only language that web browsers are able to run natively; yet JavaScript is not a language that many psycholinguistic researchers are familiar with. The first software package to facilitate this was jsPsych (De Leeuw, 2015), a dedicated JavaScript library for implementing psychological experiments; however, jsPsych is a software library rather than a graphical user interface and therefore still requires knowledge of JavaScript. More recently, it has become possible to implement online experiments through graphical user interfaces, such as OpenSesame/OSWeb (Mathôt et al., 2012, with online functionality included in 2018), PsychoPy/PsychoJS (Peirce, 2007, with online functionality included in 2017), and LabJS (Henninger, Shevchenko, Mertens, Kieslich, & Hilbig, 2021), which do not require any knowledge of JavaScript. (See the later section Alternatives for a more comprehensive overview of software packages.) These developments have made it much easier to develop online experiments.

The second limiting factor is that online experiments require a server to distribute experiments to participants in the form of URLs and to store the resulting participant data. Online-questionnaire software generally includes a server component out of the box; however, software packages such as OSWeb, PsychoJS, and LabJS do not. Initially, many researchers implemented their own ad hoc solutions for this. However, the emergence of tools such as the open-source JATOS server software (Lange, Kühn, & Filevich, 2015), especially in combination with the freely accessible MindProbe server (https://mindprobe.eu), the free-to-use Cognition.run server (https://www.cognition.run) for jsPsych experiments, and the commercial Pavlovia server (https://pavlovia.org), has made this aspect also of online experimentation considerably easier.

The third limiting factor is the widespread belief among researchers that online experiments do not offer sufficient quality control (e.g., Hamrick, in
press). After all, if you do not know who your participants are, and if you do not know under which circumstances and on what kinds of computer they are performing your experiments, then how can you trust your data?

The aim of this Methods Showcase Article is to provide a practical introduction to running behavioral experiments online, with a focus on experiments that collect reaction times and accuracy data. In doing so, we will touch upon all of the issues mentioned above. We will focus on a specific workflow that is centered around OpenSesame/OSWeb, and, using a semantic-categorization experiment that we recently conducted as a concrete example, we will address the specific challenges associated with the kinds of psycholinguistic experiment that are of special interest to the readers of Language Learning.

Workflow
Our workflow is centered around three tools: OpenSesame/OSWeb (Mathôt et al., 2012), JATOS (Lange et al., 2015), and Prolific. Each tool will be discussed in more detail below, but to make this section easier to understand, we start with an outline of the workflow:

1. The experiment is built with OpenSesame/OSWeb, which is a user-friendly tool for building experiments.
2. The experiment is exported from OpenSesame/OSWeb and imported into a JATOS server, such as MindProbe or a server that has been set up by a researcher’s own organization, which is where the experiment is hosted.
3. Prolific is used to recruit participants. Prolific directs participants to the experiment on JATOS.
4. The data are downloaded from JATOS and converted to a spreadsheet format (using a conversion tool provided with OpenSesame/OSWeb) that is convenient for data analysis.

The tools described in this section all evolve rapidly. Therefore, we will limit ourselves to those aspects of the workflow that are likely to remain valid in the near future. For implementation details, which are likely to change as tools are updated, we recommend consulting the online documentation of the tools themselves, and we provide relevant links in this article.

Designing the Experiment: OpenSesame/OSWeb
OpenSesame is a graphical experiment builder for the social sciences (Mathôt et al., 2012); it is free and open-source software that runs on Windows, MacOS, and Linux. OpenSesame provides a comprehensive graphical user interface that allows users to implement many experiments without coding. For
The OSWeb extension for OpenSesame allows the researcher to test an experiment in a browser (without having to upload it to a server first), to export the experiment to a format that can be imported into JATOS, and to check whether the experiment is compatible with OSWeb.

Additional flexibility, users can also include scripting in their experiment, using either Python or JavaScript as a programming language.

OpenSesame supports presentation of complex stimuli, both visual and auditory, and collection of many kinds of response types, including key presses, screen taps, and eye movements. OpenSesame also supports advanced randomization options, including constraints that, for example, allow researchers to limit the number of direct repetitions of a stimulus (or a condition) or to enforce a minimum distance between repetitions of a stimulus (or a condition). This functionality makes OpenSesame well suited for psycholinguistic studies. OpenSesame also provides limited support for questionnaires (forms); however, OpenSesame is not dedicated questionnaire software, and tools such as Qualtrics or SurveyMonkey are better suited to questionnaire-based experiments.

OSWeb is a JavaScript application that runs OpenSesame experiments in a browser. OSWeb supports much, but not all, of the functionality that is offered by OpenSesame for lab-based experiments. OSWeb is preinstalled as an extension in the official OpenSesame packages (Figure 1).

An important limitation of OSWeb, as compared to OpenSesame on the desktop, is that it does not support custom Python scripts (inline_script items),
which many advanced OpenSesame users rely on for developing their experiments; instead, OSWeb supports custom JavaScript (inline_javascript items). To check whether an experiment is compatible with OSWeb, users can use the “Compatibility check” that is part of the OSWeb extension. This check will highlight problems and also suggest improvements for running experiments online, such as reducing the number of variables that are logged to save bandwidth (Figure 1).

In most cases, users can simply develop their experiment using the OpenSesame desktop application and use the OSWeb “Compatibility check” and “Test experiment in external browser” functionality to make sure that the experiment runs in a browser. However, when the experiment is distributed to participants from a service such as Prolific (as in our example study), Sona Systems, or Mechanical Turk, it is often useful (though not required) to log the unique identifier (e.g., the Prolific ID) that such systems assign to the participant. With this unique identifier, the user can determine afterwards which result entry in JATOS (see below) corresponds to which participant on Prolific, Sona Systems, or Mechanical Turk. This is necessary for purposes such as contacting participants or withholding credit. To do this, an inline_javascript item is added to the start of the experiment. This script first checks whether the experiment is hosted on a JATOS server, and if information was passed from Prolific to JATOS; if so, the script retrieves the Prolific ID from this information and logs it to the OSWeb results.²

In summary, OpenSesame is a program for developing experiments. OSWeb is a JavaScript application for running OpenSesame experiments in a browser. Developing online experiments does not require any knowledge of web technologies; however, not all of OpenSesame’s functionality is available in a browser, and users should therefore check whether their experiment is compatible with OSWeb.

**Hosting the Experiment: JATOS**

JATOS (Lange et al., 2015) is open-source web-server software for managing online experiments. OpenSesame allows users to export their experiment to a format that can be imported into JATOS. Once an experiment has been imported, JATOS provides URLs, called “workers,” that can be distributed to participants (Figure 2). Participant data is also stored on JATOS. In addition to OpenSesame, JATOS also supports experiments created by a variety of other tools, such as jsPsych (De Leeuw, 2015) and lab.js (Henninger et al., 2021).

JATOS is not a single server with a single web address. Rather, it is open-source software that institutions (or individuals) can install on their own
A JATOS server generates study URLs that can be distributed to participants. The server in this screenshot is hosted at https://jatos.cogsci.nl/, but many different JATOS servers exist, typically installed by universities or research institutes on their own servers. A free JATOS server is available at https://mindprobe.eu/.

The advantage of using an institution-hosted JATOS server is that it provides institutions with maximum control over where and how their experimental data is stored, thus ensuring compliance with privacy-and-ethics requirements (see also the later section Privacy and Ethics). The disadvantage is that not all researchers have access to a JATOS server, nor the technical expertise to set up their own server. To mitigate this issue, the European Society for Cognitive Psychology (ESCOP; https://www.escop.eu/) has sponsored the launch of Mind Probe (https://mindprobe.eu), a JATOS server that is freely accessible to researchers (including non-ESCOP members).

JATOS is not a platform for recruiting participants. Rather, participants are typically recruited through another platform, such as Prolific as used in our example experiment (alternatives include Sona Systems and Mechanical Turk). The relationship between JATOS and a participant-recruitment platform such as Prolific can be confusing at first, but the logic is straightforward: Participants register for an online study on Prolific; from there, they are redirected to a URL that corresponds to an experiment on a JATOS server (Figure 2); when the experiment is finished, participants are redirected back to Prolific.

In summary, JATOS is open-source web-server software for managing experiments, generating study URLs, and storing data. Institutions (or individuals) can set up their own JATOS server to retain maximum control over where
and how their data is stored, or they can make use of a freely accessible server such as MindProbe.

**Recruiting Participants: Prolific**

Prolific (https://www.prolific.co/) is a commercial platform for recruiting participants for online experiments. When researchers create a study on Prolific, they indicate how many participants they want to recruit; how long the study takes to complete; how much each participant is paid (with a suggested rate of £7.50 or $9.60 per hour); and a wide variety of prescreening criteria. For linguistic research, relevant screening criteria can be related to the participant’s first language, other languages, bilingualism, and language-related disorders. Anyone is free to sign up as a participant on Prolific after filling out a short screening form. Prolific does not provide a systematic way to inspect demographics, but the participant pool is non-representative, self-selected, and heavily skewed toward young (< 30 years) English-speaking participants; however, given the size of the participant pool, it is possible to find a reasonable number of participants to match most criteria, including non-English native speakers and elderly people. For our example study (see Example Experiment), we recruited only native Dutch speakers.

The communication between JATOS and Prolific involves two URLs. The first is the study URL, which is a URL on a JATOS server that should be copied and pasted into the Study Link field on Prolific (Figures 2 and 3). As mentioned above, each participant has a unique Prolific ID, which is passed to the experiment as a parameter that sits within the study URL. The second URL is the end-redirect URL (Figures 3 and 4), which is a URL on Prolific that should be copied and pasted into the Study Properties on JATOS. The end-redirect URL is used to direct participants back to Prolific and to let Prolific know that a study was successfully completed for a participant.

When publishing a study on Prolific, it is prudent to first test whether everything works as expected. Prolific provides a test URL that researchers can use to test the experiment themselves from the perspective of a participant. Next, we recommend recruiting a small number of participants (e.g., five). This provides a crucial final opportunity to check whether everything runs smoothly, before recruiting a larger sample. More specifically, this allows researchers to check whether all five participants were able to successfully complete the experiment, and if not, whether this was due to their simply abandoning the task, which happens frequently and is not necessarily a cause for concern (see the Data Quality: Performance section, under Practical Considerations), or due to a
Figure 3  The study URL needs to be copied from JATOS (see Figure 2), extended with the PROLIFIC_PID, STUDY_ID, and SESSION_ID parameters, and pasted into the Study Link section on Prolific. This allows Prolific to direct participants to an experiment that is hosted on JATOS.
problem with the experiment (see the next subsection Verifying, Downloading, and Processing Data).

Once a researcher is confident that the study is running smoothly, the remaining participants can be recruited. If the prescreening criteria are not too restrictive, participant slots tend to fill up in minutes. When a participant does not complete the experiment within a specified time limit, Prolific automatically recruits a new participant. As a result, a complete dataset is generally collected within an hour; because many participants start the experiment in parallel, the duration of data collection can be independent of the size of the collected dataset.

We recommend that online experiments do not take too long to complete. This recommendation is based on the assumption that, with longer experiments, participants may be less likely to complete the task, and even if they do, they may lose interest and thus generate low-quality data. In addition, it is easy to collect many participants in online experiments, thus making it feasible, for trial-based experiments, to use a sampling strategy of many participants with a limited number of trials each. Brysbaert and Stevens (2018) recently estimated that a properly powered within-subject experiment requires around 1,600 observations per condition. (Clearly, this is no more than a rough rule of thumb, because statistical power depends strongly on effect size.) To reach this in a
In summary, Prolific is a commercial participant-recruitment platform for online experiments. The experiments themselves are hosted elsewhere, for instance on a JATOS server, as in our example experiment. Participants are first redirected from Prolific to the experiment, and upon completion are redirected back from the experiment to Prolific.

Verifying, Downloading, and Processing Data

Once the desired number of participants has successfully completed the experiment, the resulting data can be viewed on, and downloaded from, JATOS. Each entry on JATOS corresponds to one experimental session, which lasts from the moment that a participant is redirected to the experiment (hosted on a JATOS server) from Prolific, until the moment that the participant is redirected back from the experiment to Prolific (Figure 5). Sessions that have been successfully completed are marked with the state FINISHED. It is often the case that many entries have a different state, such as DATA_RETRIEVED or FAIL. These correspond to participants who started but did not complete the laboratory setting, researchers might favor a sampling strategy of 32 participants with 50 observations per condition each (32 \times 50 = 1,600); in contrast, in an online experiment, researchers might favor a sampling strategy of more participants with fewer observations per participant, such as 100 participants with only 16 observations per condition each (100 \times 16 = 1,600). Here, “condition” refers to a single level of an experimental variable, or to a combination of levels of experimental variables in a multifactorial design.
experiment, or to participants who reloaded the page (which is not allowed by JATOS), or to bugs in the experiment itself that caused the experiment to crash.

Because a substantial number of non-FINISHED entries is normal (see Data Quality), it can be difficult to spot when something is wrong; specifically, bugs in the experiment that only affect a few participants are easily overlooked, thus resulting in affected participants unfairly missing out on their payment. To minimize the risk of this happening, and as we mentioned earlier (but it is worth repeating), it is prudent to extensively test the experiment during development (see also Designing the Experiment: OpenSesame/OSWeb) and to first recruit a small sample of participants to check whether the experiment runs smoothly, before recruiting the full sample of participants (see also Recruiting Participants: Prolific).

Data from OpenSesame/OSWeb experiments are stored in a nonstandard format that cannot be read directly by commonly used software packages. To make data analysis easier, the OSWeb extension for OpenSesame provides a tool to convert JATOS results (as stored by OpenSesame/OSWeb) to a standard .csv or .xlsx spreadsheet that can be read by many commonly used software packages, such as R (R Core Team, 2014), JASP (JASP development team, 2021), and Microsoft Excel. For most trial-based experiments, this spreadsheet will be structured such that each row corresponds to a trial and each column corresponds to an experimental variable.

From this point onwards, researchers can analyze the data using their preferred analysis technique; an overview of all possibilities is beyond the scope of this article, but we will provide three common examples: First, researchers might create a pivot table in Excel and perform a (Bayesian) repeated-measures ANOVA in JASP; second, they might load the data into R and perform a linear mixed-effects analysis with lme4 (Baayen, 2008; see Gries, 2021); third, they might load the data into Python and perform a linear mixed-effects analysis with statsmodels (Seabold & Perktold, 2010). As we will discuss in the Example experiment, we have taken the third approach.

Practical Considerations
There are numerous practical issues to consider when building an online OpenSesame/OSWeb experiment. Here we will focus on issues related to temporal precision and accuracy, counterbalancing, and the extent to which researchers can verify whether participants are performing the experiment seriously.
Data Quality: Temporal Precision and Accuracy
An often-voiced concern regarding online experiments is related to timing: How accurately and precisely can researchers control stimulus-presentation timing? And how accurately and precisely can they measure response timing? Here, “precision” refers to the variability of a measurement (one kind of reliability), whereas “accuracy” refers to how close, on average, a measurement is to the true value (one kind of validity).

Recent benchmark studies have measured the temporal precision of visual-stimulus presentation with different packages for online experiments (Anwyl-Irvine, Dalmaiier, Hodges, & Evershed, 2020; Bridges, Pitiot, MacAskill, & Peirce, 2020; Kuroki, 2021). (There are also a number of older benchmark studies, e.g., Pinet et al., 2017; Reimers & Stewart, 2014. However, given the rapid developments of browser technologies, these studies are likely no longer representative.) Key metrics are the standard deviation of display durations (precision) and the mean deviation of the actual display durations from the intended display durations (accuracy). The picture that emerges from these studies is that temporal precision is decent, with some studies reporting a standard deviation of less than 5 ms (Bridges et al., 2020) and others reporting a slightly higher standard deviation of around 10 ms (Anwyl-Irvine et al., 2020). Temporal accuracy is also decent, again with some studies reporting mean lags of around 5 ms (Bridges et al., 2020) and others reporting mean lags of around 15 ms (Anwyl-Irvine et al., 2020). At the time of writing, the best temporal precision appears to be provided by jspsych-psychophysics (Kuroki, 2021), a plugin for jsPsych (De Leeuw, 2015), which claims standard deviations of less than 1 ms under some conditions, approaching the near-perfect temporal precision that can be obtained in traditional laboratory setups. In all cases, there is considerable variability between operating systems, browsers, and software packages; therefore, the values reported above are merely intended to convey a rough picture of the kind of precision and accuracy that one can realistically expect under optimal circumstances.

A key term in the preceding sentence is “optimal circumstances.” The problem with timing in online experiments is not primarily that browsers are unable to offer near-perfect timing on optimal systems; rather, the problem is that most participants are not using optimal systems, which necessarily leads to less-than-perfect timing. A crucial aspect of designing a successful online experiment is therefore to choose a design that is robust to less-than-perfect timing. Fortunately, and contrary to what many researchers believe, this robustness holds for the vast majority of (behavioral) psycholinguistic studies.
Let us first consider our example experiment (see Example experiment) in which participants saw a string of one or two words and then pressed a key on the keyboard to indicate whether this string was related to animals or not. How important is temporal accuracy in this case? Not very important at all, because we are not interested in absolute reaction times (RTs; remember that accuracy refers to the mean deviation of the measured value from the true value), but rather in how RTs are affected by our (within-subject) manipulation. And how important is temporal precision? Precision is important in the sense that decreased precision will lead to decreased statistical power (remember that precision refers to variability, or noisiness, of the measurement); that is, a decreased sensitivity in detecting an effect of our manipulation. Therefore, less-than-perfect precision is somewhat of a problem. But one should not exaggerate this problem either, because the reduction in statistical power due to less-than-perfect precision is limited; human RTs are inherently variable (with a standard deviation of around 150 ms in most tasks), and therefore a small measurement error increases this natural variability only slightly (Avery & Marsden, 2019; Damian, 2010; Vadillo & Garaizar, 2016).

However, let us now consider a masked-priming experiment, such as conducted by Petit, Midgley, Holcomb, and Grainger (2006), in which a single-letter prime is presented for 33.3 ms (two frames on a typical 60-Hz display), preceded and followed by a mask (a “#” character). Here, the exact presentation duration of the prime is crucial: If the prime is shown for one frame less (16.7 ms), then it may be presented too briefly to have a measurable priming effect; if the prime is shown for two frames less then it is not shown at all; and if the prime is shown for too many frames, then it will be clearly visible, and participants will be acutely aware of the priming manipulation. Therefore, in this case, it is crucial to have both perfect temporal accuracy and precision, because the prime should be shown for exactly 33.3 ms on every trial. If not, this would not merely reduce the statistical power of the study, but invalidate its entire premise.

OpenSesame/OSWeb automatically stores timestamps for relevant events, such as the onset of a stimulus display, which is handled by a sketchpad. Based on these timestamps, and assuming there is at least one stimulus in the experiment with a fixed duration, researchers can obtain an estimate of whether the actual presentation durations of stimuli match the intended durations. As discussed under Example Experiment, in our example study we verified the presentation duration of the fixation dot to obtain a rough estimate of the temporal precision of our participants’ systems. Importantly, these timestamps are based on “introspection” by the browser and are reliable only insofar as the browser
is able to provide accurate temporal information. Therefore, these timestamps are useful for detecting extremely poor temporal precision, which in our experience is rare but does occur; however, they cannot be used to verify that temporal precision was perfect.

In summary, when running an experiment online, temporal accuracy and precision should be considered as probably decent but essentially unknown. This is not primarily because browsers are unable to offer near-perfect timing on optimal systems, but rather because most participants are not using optimal systems. For most experimental designs, imperfect timing merely reduces statistical power. However, for some experimental designs, imperfect timing poses a threat to the validity of the study. As a researcher, it is important to make sure that the experimental design is robust to imperfect timing before deciding to run it online.

Counterbalancing

“Counterbalancing,” as the term is most often used, means that some variable in an experiment, such as response mapping, is systematically changed between participants to avoid it being a confound; for example, in a lexical-decision experiment, some participants may press the right arrow key to indicate that a word was presented and the left arrow key to indicate a nonword, whereas other participants press left for words and right for nonwords. Counterbalancing is distinct from variables that are changed (systematically or randomly) within a single experimental session, such as the order of trials within a block. Importantly, whereas OpenSesame/OSWeb offers extensive support for changing variables within experimental sessions,\(^4\) between-session counterbalancing of online experiments requires some consideration.

In our experiment, participants saw strings that corresponded to negations (e.g., “no fish”) or positive assertions (e.g., “jacket”), which was varied between blocks, rather than within blocks; that is, participants either first saw an entire block of negations followed by a block of positive assertions, or the other way around. In this counterbalancing example, half the participants started with negations, whereas the other half started with positive assertions, so that the effect of negation was not confounded by block order.

In an offline task, counterbalancing is often linked to the participant number, such that, for example, all even-numbered participants (0, 2, …) start with negations and all odd-numbered participants (1, 3, …) start with positive assertions. When running an experiment online, this approach is complicated by the fact that most systems assign a random unique identifier to each participant or each session. For example, JATOS (see Workflow) assigns a jatosResultId that
is unique to each experimental session, and Prolific (see Workflow) assigns a PROLIFIC_PID that is unique to each participant but constant across sessions. Importantly, neither of these unique identifiers are linearly incrementing numbers that could be used for counterbalancing.

To work around this problem, a researcher can randomly assign a block order to each participant and accept the risk that there may be slightly more participants starting with the negations than with the positive assertions. This is the approach that we took in our example experiment, and by chance we ended up with only 20 participants who started with the negations versus 29 who started with the positive assertions—clearly an imperfectly balanced result.

As a straightforward alternative, researchers can create two versions of the task (or more if the counterbalancing rule is more complicated) and manually distribute each version to an equal number of participants to achieve perfect counterbalancing.

As a more elegant and flexible alternative, but one that requires more technical skill, researchers can also make use of so-called Batch Session Data in JATOS. Batch Session Data is a data pool that is shared between all sessions of the same “batch,” where a batch generally corresponds to a single study. For our example experiment (if we had chosen this route), the Batch Session Data could have contained two numbers: one that indicates how many participants are still required for the negation-first block order and one that indicates how many participants are still required for the positive-assertions-first block order. Whenever a new session is started (i.e., when a participant clicks on the experiment URL), the experiment checks the Batch Session Data to see which order requires the most participants and then assigns this order to the current session. When both counters reach zero, the experiment stops, with a message informing the participant that the experiment has already been completed, and that he or she can therefore no longer participate. (Ideally, of course, the researcher recruits exactly the required number of participants so that this does not happen.) The advantage of using Batch Session Data is that it offers considerable flexibility and also allows experimental sessions to have “memory,” such that counterbalanced designs can span across multiple sessions. For example, Zhou, Lorist, and Mathôt (in press) used Batch Session Data to implement a design in which participants completed four separate experimental sessions while counterbalancing the order of these sessions between participants.

Data Quality: Performance

Another often-voiced concern regarding online experiments is related to data quality: How can researchers be sure that participants are taking an
experiment seriously when they are running it by themselves on their own computers?

In our experience, the dropout rate of participants in online experiments is far higher, in the order of 10–20%, than in lab-based experiments, where the dropout rate is close to 0%; in other words, many participants start an online experiment but never finish it, whereas this rarely happens in the lab. In this sense, the concentration of useable data is relatively low. But given that researchers can simply recruit participants until they have a sufficient (and ideally predetermined; Wagenmakers et al., 2012) number of complete datasets, a high dropout rate is in itself not a problem. A more important question is: How valid and reliable are the data for those participants who do finish the experiment? To assess the quality of complete and valid datasets, it is important to have some measure in the experiment that is objectively good or poor.

Let us first consider our example experiment, in which we measured RTs and response accuracy, which naturally lend themselves to assessing the quality of the data. As discussed below under Example experiment, this allowed us to exclude participants for whom RTs and/or response accuracy are outside of the “normal range,” where the normal range is often defined as two standard deviations around the grand mean RT or response accuracy (though see Marsden, Thompson, & Plonsky, 2018, showing the wide variability in outlier identification practices). This approach is not specific to online experiments; however, to the extent that data from online experiments tend to be more variable than data collected in laboratory settings, data-quality checks are even more important.

However, assessing data quality is not always as straightforward as in our example experiment. Consider an online experiment in which normative ratings are collected for a set of linguistic stimuli. Normative ratings do not, or least not necessarily, allow for a straightforward assessment of data quality: How can you tell whether a participant was taking the experiment seriously based on, say, how much they report the word sun as having a positive valence (Mathôt, Grainger, & Strijkers, 2017)? In such experimental designs, it is possible to include a secondary task (an “attention check”) that requires a certain amount of attention from participants in order for them to perform well, but that does not interfere with the task of interest (Kung, Kwok, & Brown, 2018). This secondary task can be as simple as occasionally presenting either a triangle or a square, and having participants quickly press a key when they see a triangle, but withhold a key press when they see a square. Participants can then be excluded when they respond too slowly (or not at all) on this task, and/or when they have a large number of false alarms (i.e., when they press a button when
a square was presented). Attention checks are a useful way to verify that participants are performing a task seriously; however, they may also change how participants interpret and perform a task (Hauser & Schwarz, 2015). Therefore, researchers should always critically assess whether attention checks are necessary in a specific experiment.

In summary, although many participants who start an online experiment do not finish it, this is not necessarily a problem as long as data quality for those participants who do finish is decent. In order to assess data quality, it is important to include some measure in the experiment, such as RT or response accuracy, that indicates whether participants performed the task as intended.

**Privacy and Ethics**

Online experiments rely on a complex digital infrastructure, and data from online experiments are often stored in the cloud on physical servers whose locations are unclear to researchers as well as participants. Importantly, data that is stored on third-party servers may be subject to regulations that are similarly unclear to researchers and participants. To complicate matters further, universities and research institutes often impose restrictions on where and what kind of research data can be stored. Given these complexities, how can researchers conduct online experiments while respecting privacy-and-ethics regulations?

In terms of data content, as little personally identifying information about participants as possible should be stored (Klein et al., 2018). However, even if an online experiment does not collect any personal information, the data still contains unique personal identifiers, such as the Prolific ID (see the earlier Workflow section). This means that the raw data from online experiments are rarely, if ever, fully anonymous. Participants should be informed about this and actively consent to their data being used. This consent can be sought, for example, by presenting a digital informed-consent form at the start of the experiment; this form should require a nontrivial action from participants to indicate consent and should not allow participants to start the experiment before they have done so. OpenSesame allows users building the experiment to insert a form_consent item into the experiment for this purpose. When publicly sharing data from online experiments, researchers should remove personal identifiers; for the data from our example experiment (Mathôt & March, 2022d), we did this by replacing each Prolific ID by a randomly chosen number.

In terms of data storage, the optimal solution is to install JATOS on servers that are maintained by universities or research institutes themselves; this makes it entirely clear where data are stored (at the university or research institute) and to whom they belong (to the university or research institute). The downside of
this approach is that it requires a technical department that is willing and able to set up and maintain such a server.

If an institutional JATOS server is not available, researchers can also use third-party servers for hosting online experiments. When doing so, it is important to review the ethics-and-privacy policy of the server and, when in doubt, consult a privacy officer (nowadays employed by most universities and research institutes). It is also preferable to use a server that runs open-source software, which allows for maximum transparency. MindProbe runs open-source software (JATOS); its privacy-and-ethics policy is intended to protect the rights of researchers and participants and is occasionally updated based on feedback from researchers and privacy officers. However, because privacy regulations vary between countries and institutions, the responsibility of ensuring that data storage complies with all regulations ultimately lies with the researcher.

Example experiment
We investigated whether word recognition is facilitated for words that are associated with brightness when they are presented in a bright font (e.g., “sun” presented in a bright font), and for words associated with darkness when they are presented in a dark font (e.g. “night” printed in a dark font). In other words, we asked whether word comprehension is facilitated when the semantic and sensory properties of a word are congruent; this prediction was based on embodied views of language, which posit that word meaning is in part derived from sensory and motoric representations (Glenberg & Gallese, 2012; Pulvermüller, 2013). We further asked whether the predicted congruency effect would invert for words that are negated (e.g. interference for “no sun” printed in bright font). Finally, we aimed to replicate the effect of negation (e.g. Dudschig & Kaup, 2018; Kaup & Dudschig, 2020) such that participants should respond more slowly to negated words (“no sun”) compared to positive assertions (“sun”).

To investigate these questions, participants performed a semantic-categorization task in which they indicated whether words were related to animals or not, a task that was unrelated to our research questions, but did require semantic processing. Similar experimental paradigms have been used to investigate language learning, for example by comparing different age groups (Colé et al., 1999; Grainger et al., 2012) or populations (Martens & de Jong, 2006), or by comparing the second and first language in bilinguals (Keatley & Gelder, 1992).

Our experimental scripts (Mathôt & March, 2022a), analysis code (Mathôt & March, 2022b), coding scheme (Mathôt & March, 2022c), and data (Mathôt
& March, 2022d) have all been made publicly available via IRIS (https://www.iris-database.org) and the Open Science Framework (https://osf.io/qnt93/?view_only=cf930630a087406fa16961b65182009a).

Methods
The experiment was conducted following the general workflow outlined above: first, the experiment was set up in OSWeb; second, it was hosted on JATOS; and third, Prolific was used to recruit participants. The experiment was approved by the ethics committee of the University of Groningen (PSY-1920-S-0408).

First, the experiment was implemented in OSWeb (see Designing the experiment: OpenSesame/OSWeb). A detailed overview of the experimental design can be found in March (2020). In short, we investigated the effect of word type, negation, and font brightness on response time. Word type included darkness-related words (e.g., night), brightness-related words (e.g., sun), animal-related words (e.g., fish), and other/control words (e.g., jacket; adapted from Mathôt et al., 2019). Half the trials included negations and the other half did not (see Counterbalancing). We varied font brightness randomly across stimuli (i.e., stimuli were presented in either black or white font).

On each trial, participants performed a semantic-categorization task. Specifically, participants saw a string consisting of one or two words, and indicated whether this string was associated with animals or not. The response-key mapping was randomly selected for each participant (i.e., randomly counterbalanced), such that some participants pressed the left arrow key for animal-associated strings and the right arrow key for other strings, while other participants pressed the right arrow key for animal-associated strings and the left arrow key for other strings.

Second, the experiment was uploaded on JATOS and connected to the recruitment platform Prolific via a redirection URL (see Hosting the experiment: JATOS).

Third, using Prolific’s screening criteria, we recruited only Dutch native speakers and did not allow participants to sign up more than once. After participants finished the experiment hosted on JATOS, they were redirected to Prolific via a second URL (see Recruiting participants: Prolific). Finally, data was verified, downloaded, and processed as described below.

Results
The Importance of Predetermined Criteria
In the following sections, we show how to perform various steps to “clean” the data from online experiments based on various criteria. These steps offer
considerable degrees of freedom for researchers to steer their results in a particular direction. It is therefore good scientific practice to specify the exact steps that will be performed before data collection has taken place, for example by preregistering the analysis plan (Wagenmakers et al., 2012) or by following the Registered Report article route (Marsden, Morgan-Short, Trofimovich, & Ellis 2018), and to diverge from these steps after the fact only for good reasons, which should be transparent to and verifiable by peer reviewers. The following sections are intended to provide researchers with guidelines as to which steps can be taken and how.

**Complete and Valid Datasets**

Of 63 participants who signed up through Prolific, 51 participants completed the experiment. Of these 51 participants, one was excluded immediately because that participant had completed the experiment within half the time of the other participants, and an ad hoc analysis showed that this participant had been responding randomly. Another participant was excluded immediately because a technical issue occurred that caused part of the experiment to be repeated. This left 49 complete and valid datasets for further analysis.

**Data Quality: Temporal Precision and Accuracy**

The information that follows in this section is by necessity technical and involves details of how computer monitors are periodically refreshed. It is, however, important information for researchers who are interested in presenting visual stimuli with millisecond precision in online experiments. Readers for whom this is not a primary interest could skip to the next section, Data Quality: Performance.

We assessed temporal precision by checking whether the actual presentation duration of the fixation dot, as logged by the browser, matched the specified presentation duration of 500 ms. Figure 6(a) shows a histogram of the presentation durations of the fixation dot, measured as the timestamp of the target display minus the timestamp of the fixation-dot display as logged automatically by OSWeb.

Within the 490–540-ms range shown in Figure 6(a), the presentation duration of the fixation dot was usually somewhere in between 500 ms and 516 ms, with a clear peak just short of 516 ms. Assuming a monitor with a 60-Hz refresh rate, 516 ms corresponds to exactly 31 frames. (A computer monitor is refreshed at fixed intervals, usually of 16 ms. One such refresh is called a “frame.”) Crucially, the fact that the presentation duration was clustered around 31 frames, as opposed to being more or less uniformly distributed,
Figure 6 (a) The actual duration in ms of the fixation dot as logged by OSWeb based on browser timestamps and (b) the mean actual duration in ms of the fixation dot for each individual participant. Dotted lines indicate display durations that are compatible with a 60-Hz monitor (i.e., 500 ms, 516 ms, 533 ms).

suggests that on many of our participants’ systems the so-called “synchronization to the vertical refresh” or “v-sync” was enabled; that is, the onset of a new visual stimulus coincided with the start of a new refresh cycle (frame) of the monitor, resulting in display durations that are multiples of the 16-ms frame duration. A blocking flip is generally considered optimal, because if a stimulus is presented in the middle of a refresh cycle (which happens if v-sync is not enabled), there will be a short moment during which only half the monitor shows the new stimulus, which results in a visual artifact called “tearing” that is characterized by horizontal lines that seem to run across the monitor.

The fact that display durations are centered around 516 ms (31 frames), rather than 500 ms (30 frames), highlights the rule of thumb that, in OpenSesame as well as in many other software packages, the user should always specify a display duration that is slightly below the intended display duration. For example, a user could specify a duration of 495 ms for a stimulus that should be presented for 500 ms. The rationale behind this is that, due to the discrete refresh cycles of the monitor, a presentation duration of 495 ms (or 29.7 frames) is impossible and will therefore be rounded up to the next possible frame, resulting in a presentation duration of 500 ms (30 frames). In our example study, we did not do this, and therefore even the slightest delay resulted in a frame being skipped, thus explaining the peak around 516 ms (31 frames) as opposed to 500 ms (30 frames).

Finally, the fact that Figure 6(a), despite having a clear peak around 516 ms, shows considerable variability reflects that it is not realistic to expect
single-frame temporal precision from online experiments that are conducted in uncontrolled environments.

In total, only 2% of trials did not fall within the 490–540-ms range shown in Figure 6(a). Figure 6(b) shows the mean presentation duration for all 49 valid participants, and clearly highlights a single participant as having exceedingly high presentation durations. Quite possibly, this participant was running the experiment on a low-performance system, or with many other programs running at the same time. This one participant accounted for all but one of the deviant presentation durations (i.e., those falling outside of the range shown in Figure 6(a); one other participant had a single slightly deviant duration of 570 ms, which we did not exclude). Therefore, we excluded this participant, leaving 48 complete, valid, and temporally precise datasets for further analysis.

**Data Quality: Performance**

Figure 7 shows the mean RT (across both correct and incorrect trials) and accuracy for each of the 48 remaining participants. Most participants have RTs between 500 and 900 ms, and accuracies of 70% or higher, which, based on our experience with similar experiments, is reasonable. We excluded participants with a mean RT that deviated by more than two standard deviations from the grand mean RT (i.e., the mean of the per-participant mean RTs), and
participants with an accuracy that deviated by more than two standard deviations from the grand mean accuracy. Based on this criterion, which is often used for studies of this kind (and which, again, should ideally be predetermined), we excluded six participants, leaving 42 participants (13,524 trials in total) with, at least by this standard, high-quality data.

Next, we conducted a linear mixed-effects analysis with response times (correct trials only) as dependent measure, negation (negation vs. positive assertion) as fixed effect, and by-participant random intercepts and slopes. This revealed an effect such that negations were responded to more slowly than positive assertions.

**Alternative Workflows for Online Experiments**

The workflow that we have described in this article is centered around OpenSesame/OSWeb, JATOS, and Prolific. However, there are many other tools that researchers can use to run experiments online, and the choice of one set of tools over another is largely one of personal preference. The different tools can be combined in various ways, resulting in a large number of possible workflows, which have recently been reviewed by Sauter, Draschkow, and Mack (2020) and Grootswagers (2020). In very general terms, there are three types of tools for online experiments (commercial tools are marked with an asterisk, *):

- **an experiment builder**, where alternatives for OpenSesame/OSWeb include jsPsych (De Leeuw, 2015), lab.js (Henninger et al., 2021), PsychoPy/PsychoJS (Peirce, 2007), PsychoToolkit (Stoet, 2010), SimplePhy (Lago, 2021), and Tatool Web (https://tatool-web.com);
- **a server to host experiments**, where alternatives for JATOS include Pavlovia* (https://www.pavlovia.org/), Open Lab (https://open-lab.online/), and Cognition.run (https://cognition.run); and

A further alternative is to use an integrated service that provides all or most of the above, where options include the following:

- Gorilla* (does not provide participant recruitment; https://gorilla.sc),
- Inquisit Web* (https://www.millisecond.com/),
- Labvanced* (https://www.labvanced.com/),
- Resultal* (https://resultal.com), and
Testable* (https://www.testable.org/).

At the time of writing, the combination of OpenSesame/OSWeb and JATOS is the only free and open-source workflow that allows researchers to use a graphical user interface to build and deploy online experiments that rely on complex stimuli and collection of RT and accuracy data. However, it is not the only workflow available, nor is it necessarily the optimal workflow in every situation. Below we briefly describe two main alternative workflows, both of which are currently widely used by researchers and are compatible with most privacy-and-ethics regulations (see the earlier section Privacy and Ethics).

The first alternative workflow combines PsychoPy/PsychoJS (as a graphical user interface for building experiments) and Pavlovia (as a server for hosting experiments). This combination of tools offers similar functionality to the combination of OpenSesame/OSWeb and JATOS; specifically, PsychoPy/PsychoJS also offers a comprehensive graphical user interface for building complex experiments. However, Pavlovia runs proprietary software that cannot be installed on institutional servers and charges a (modest) fee for each participant tested or for a yearly institutional license.

A second alternative workflow combines jsPsych (as a JavaScript library for building experiments) and Cognition.run (as a server for hosting experiments). This combination of tools is well suited to researchers who prefer to code their experiments directly in JavaScript using jsPsych. However, Cognition.run, although free of charge, also runs closed-source software that cannot be installed on institutional servers.

Unfortunately, at present all large-scale participant-recruitment platforms are proprietary and charge a fee. This means that all of the three workflows described above are generally combined with a commercial service such as Prolific, Mechanical Turk, or Sona Systems for the purpose of participant recruitment.

Conclusion
We have provided a practical introduction to running experiments online, with a focus on linguistic experiments that collect reaction times and accuracy. Our workflow, which is one of many possible workflows (see, e.g., Sauter et al., 2020), is centered around the use of three tools: OpenSesame/OSWeb for developing the experiment; JATOS for hosting the experiment; and Prolific for recruiting participants. We have reviewed, and illustrated through an example study, several challenges associated with online experiments, related to timing,
counterbalancing, data quality, and ethics and privacy. In our experience, all of these challenges are surmountable, but it is important for researchers to be aware of them before deciding to run an experiment online. In conclusion, online behavioral experiments are a viable alternative to laboratory experiments for many types of psycholinguistic research.

Final revised version accepted 19 February 2022

Open Research Badges

This article has earned Open Data and Open Materials badges for making publicly available the digitally-shareable data and the components of the research methods needed to reproduce the reported procedure and results. All data and materials that the authors have used and have the right to share are available at https://osf.io/ywbej/ and http://www.iris-database.org. All proprietary materials have been precisely identified in the manuscript.

Notes

1 As a technical aside, there are two main alternatives to using JavaScript. First, browser extensions such as Adobe Flash and Java historically allowed web browsers to execute code written in languages other than JavaScript; however, such extensions are no longer supported by modern web browsers. Second, it is possible to “transpile” one language, such as Python, into another language, such as JavaScript. This technique is used in some ways by both OpenSesame/OSWeb and PsychoPy, and potentially allows online experiments to be coded in Python, a language that is more familiar than JavaScript to many researchers; however, due to fundamental differences in how languages work, transpiling is limited and error-prone. In summary, although alternatives for using JavaScript do exist, currently none of them are sufficiently well-developed and well-maintained to be a viable option for fully replacing JavaScript in online experiments.

2 For an explanation of how to use OSWeb/ JATOS in combination with Prolific, see https://osdoc.cogsci.nl/manual/osweb/prolific.

3 This information is based on our trying out different screening criteria to see how many participants match.

4 See https://osdoc.cogsci.nl/manual/structure/loop/.


References


JASP development team. (2021). JASP (0.16) [Computer software]. https://jasp-stats.org/


online studies. *PloS One*, 10(6), e0130834. https://doi.org/10.1371/journal.pone.0130834


Mathôt, S., & March, J. (2022c). *Coding scheme. Materials from “Conducting linguistic experiments online with OpenSesame and OSWeb”* [Coding scheme]. IRIS Database, University of York, UK. https://doi.org/10.48316/jvr6-3q75


and Language Data with Amazon’s Mechanical Turk, 122–130.
https://doi.org/10.5555/1866696.1866715

Neuroscience Methods, 162(1–2), 8–13.
https://doi.org/10.1016/j.jneumeth.2006.11.017

Petit, J.-P., Midgley, K. J., Holcomb, P. J., & Grainger, J. (2006). On the time course of
letter perception: A masked priming ERP investigation. Psychonomic Bulletin &

Measuring sequences of keystrokes with jsPsych: Reliability of response times and
interkeystroke intervals. Behavior Research Methods, 49(3), 1163–1176.
https://doi.org/10.3758/s13428-016-0776-3

embodied and abstract-symbolic semantics. Trends in Cognitive Sciences, 17(9),
458–470. https://doi.org/10.1016/j.tics.2013.06.004

R Core Team. (2014). R: A language and environment for statistical computing (4.1.2)
https://www.R-project.org/

Reimers, S., & Stewart, N. (2014). Presentation and response timing accuracy in
Adobe Flash and HTML5/JavaScript Web experiments. Behavior Research
Methods, 47(2), 309–327. https://doi.org/10.3758/s13428-014-0471-1

brief introduction to running behavioral experiments online. Brain Sciences, 10(4),
251. https://doi.org/10.3390/brainsci10040251


https://doi.org/10.3758/BRM.42.4.1096

parameter recovery from reaction times. BMC Bioinformatics, 17, 147.
https://doi.org/10.1186/s12859-016-0993-x

language production research with overt articulation: Proof of concept, challenges,
and practical advice. Behavior Research Methods.
https://doi.org/10.3758/s13428-021-01686-3

Wagenmakers, E.-J., Wetzels, R., Borsboom, D., Maas, H. L. J. van der, & Kievit, R.
A. (2012). An agenda for purely confirmatory research. Perspectives on


**Appendix: Accessible Summary (also publicly available at https://oasis-database.org)**

**An example workflow for conducting (reaction time) experiments online**

*What This Research Was About and Why It Is Important*

This article provides an example workflow for conducting linguistic experiments online, with a focus on the presentation of complex stimuli and the measurement of reaction times. This method allows researchers to flexibly collect large amounts of data in short periods of time. The workflow is centered around three tools: OpenSesame/OSWeb (open source) provides a user-friendly graphical interface for developing experiments; JATOS (open source) is server software for hosting experiments; and Prolific (commercial) is a platform for recruiting participants. These three tools integrate well with each other and together provide a workflow that requires little technical expertise. A semantic congruency paradigm is used to demonstrate the outlined workflow, illustrating how to deal with challenges associated with running experiments online. These include the implementation of counterbalancing and data-quality concerns regarding temporal precision and performance. Finally, the authors discuss privacy and ethical considerations.

*What the Researchers Did*

The researchers provide a workflow for conducting linguistic experiments online:

- The experiment is built with OpenSesame/OSWeb.
- The experiment is exported from OpenSesame/OSWeb and imported into a JATOS server.
- Prolific is used to recruit participants and direct them to the experiment on JATOS.
- The data from the participants are stored on JATOS and can be downloaded and converted to a spreadsheet format (using a conversion tool provided with OpenSesame/OSWeb) that is convenient for data analysis.
What the Researchers Found

- A semantic congruency paradigm using the outlined workflow illustrates how the presentation of complex stimuli and the collection of reaction time data can be implemented online.
- Even though temporal accuracy (whether the measurement of reaction times is valid) and precision (whether the measurements of reaction times are stable over time) are imperfect in online experiments, this does not necessarily pose a threat to a study’s validity.
- Even though dropout rates in online experiments are higher than in traditional laboratory experiments, data quality of participants is reasonable and recruitment easier.

Things to Consider

- Some experiments cannot be implemented online (for example, experiments in which participants are small children, additional measures such as eye-tracking are being collected, or temporal precision is crucial).
- Counterbalancing can be implemented online, but this requires careful planning.
- Online experiments rely on a complex digital infrastructure, and privacy and ethics policies differ between countries and institutes. Therefore, researchers should consult with their institutes before running experiments online.
- The outlined workflow is one of many options for running experiments online.

Materials, data, open access article: Materials and data are publicly available at IRIS (www.iris-database.org) and OSF (https://osf.io/ywbej/).

How to cite this summary: March, J., & Mathôt, S. (2022). An example workflow for conducting (reaction time) experiments online. OASIS Summary of Mathôt & March (2022) in Language Learning. https://oasis-database.org

This summary has a CC BY-NC-SA license.