Gallant, J. & Libben, G. (2019). No lab, no problem: designing lexical comprehension and production experiments using PsychoPy3. *The Mental Lexicon,* 14(1), 152–168

No lab, no problem: Designing lexical comprehension and production experiments using PsychoPy3

Jordan Gallant & Gary Libben

Brock University, Canada

Abstract: We present new opportunities for psycholinguistic research that are made available by presenting experiments online over the web. We focus on PsychoPy3, which is a new version of a system for the development and delivery of behavioural experiments. Crucially, it allows for both these functions to be performed online. We note that experiments delivered over the web have significant efficiency advantages. They also open up new opportunities to increase the ecological validity of experiments and to facilitate the participation of members of populations that have thus far been less studied in the psycholinguistic literature. We discuss the crucial matter of millisecond timing in online experiments. The technical details of implementation of a behavioural psycholinguistic experiment are presented, along with listings of additional technical resources and support. Our overall evaluation is that although online experimentation still has technical challenges and improvements are ongoing, it may well represent the future of behavioural psycholinguistic research.

At the most fundamental level, a behavioural psycholinguistic experiment requires that three conditions be fulfilled: (a) the researcher must be able to recruit participants engaging in language processing, (b) the researcher must be able to access tools capable of measuring those language processes, and (c) the researcher must be able to bring the participants and the tools together.

Traditionally, satisfying condition (c) has required that the measurement tools and the participants be in the same place. Psycholinguistic laboratories are typically located in university environments. It is therefore not surprising that so much behavioural psycholinguistic research has relied on the recruitment of undergraduate university students, who constitute the most numerous group in such environments. It is also not surprising that other very large groups (e.g., persons who have not received post-secondary education, persons over the age of 30) have been considerably less studied.

There is no doubt that the investigation of language processing in university-based psycholinguistic laboratories has led to significant advances in knowledge. The laboratories are designed to be optimally controlled environments. University students, as participants, are highly skilled language users and highly disciplined test takers. Nevertheless, it seems safe to expect that there is some associated loss of sampling breadth. For example, an analysis of the participant selection patterns of the last five years (2014-2018) of this journal yields what we expect is a typical picture: Of the 58 studies reported that involved adult participants performing a language processing task, 32 of those (55%) solely involved university students.

Correspondence: Jordan Gallant jg09eh@brocku.ca Gary Libben glibben@brocku.ca
The development of this research was supported by a DGS Research Fellowship from the Faculty of Graduate Studies at Brock University and by Social Sciences and Humanities Research Council of Canada Partnership Grant 895-2016-1008 (“Words in the World”).

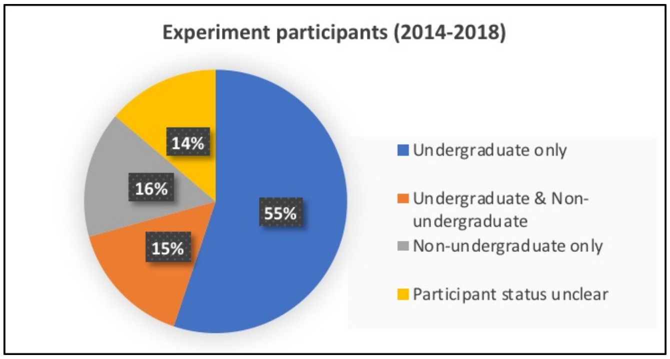

Figure 1. In the five-year period (2014-2018), 58 articles in *The Mental Lexicon* reported language processing experiments with adult participants. This represents the tabulation of how many of those focused on undergraduate and non-undergraduate participants.

There is every reason to believe that undergraduate university students will continue to play an important role as participants in psycholinguistic experiments. In many cases their participation is a key part of their education and part of their acculturation into the research environment of their universities. As we have mentioned, they typically have very high levels of language processing skills. Nevertheless, it seems to us that the recent availability of online tools for psycholinguistic experimentation could serve to change the research landscape very dramatically.

Online experiments can change the psycholinguistics research landscape

We use the term 'online experiments' to refer to behavioural psycholinguistic experiments that are conducted without a psycholinguistic laboratory. Instead of booking an appointment to enter a physical laboratory room to use a computer that is pre-loaded with experimental software, stimuli, etc., the participant in an online experiment typically uses his or her own computer to run the experiment. The participant accesses the experimental software and stimuli through a link that is provided by the researcher. The participant's data are typically uploaded to an online repository for analysis by the researcher or the research team.

As the description above suggests, online experimentation affects the activities of both the experimenter and the participants. It also may affect the overall character of the research enterprise. Online experimentation can impact the ecological validity of research and perhaps open up entirely new lines of research. It thus has real potential to change the research landscape in psycholinguistics.

Experimenter impact: inclusivity, cost effectiveness, time efficiency, and collaboration

Online experimentation does not have the same barriers to entry as other lab-based methods of inquiry. Anyone with the skills and interest can become a member of the community of psycholinguistic experimenters, irrespective of whether they have, or have access to, a physical psycholinguistics laboratory. For instance, students and other early researchers who may not be in possession of a research grant can conduct experiments with a large number of participants. Even in a funded laboratory setting, online experimentation allows researchers to spend fewer hours testing participants and more hours engaged in experiment generation, data analysis, and data interpretation. Additionally, the simple fact that online experimentation is less expensive can encourage the pursuit of riskier research topics for both students and more senior researchers.

Through the collaborative experiment development framework enabled by platforms such as Gitlab and Pavlovia (discussed below), experiment developers can, for example, insert their particular stimulus sets into existing experiments. They can share segments of experiments (e.g., particular production or recognition tasks, rating tasks, etc.) with others and they can collaborate to create joint experiments or parallel cross-linguistic studies. It therefore, offers an in-road into experimental work for scholars in related fields of language research (e.g., those that are more humanities based).

Participant impact: inclusivity, empowerment, partnership

Online experimentation makes it much easier to bring a greater number and diversity of participants into the psycholinguistic research enterprise. Even if the participant population would remain restricted to undergraduate students, it is possible for a great many more to participate in a single day of experimentation than would be possible in laboratory-based experimentation.

The real power of the approach, though, is that it enables participants to be tested who might not have easy access to a university location or who are members of a specific participant group (defined, for example by age, language type of bilingualism, level of education, or geographic location). The use of online experiments ensures that if members of such groups can be recruited over the web, they can also be tested over the web. It is possible that the over-reliance of convenience sampling from university student populations, necessitated by in-lab experimentation has coloured our perspective on language processing (Speed, Wnuk & Majid, 2017). Moving experiments online reduces geographic restriction on sampling so that more diverse populations can be reached and studied, even with a small research budget. This may also have the effect of facilitating international collaboration in the construction and design of experiments, and also in participant recruitment.

Online experimentation may have participant empowerment effects by increasing participant agency and autonomy. Online experiments can be completed at any time. For behavioral experiments not requiring specialized equipment, they can be run in an environment selected by the participant themselves. This is true of all participants. Despite decades of improvement through protocols developed for the ethical conduct of research, laboratory conditions can instill a sense of power imbalance between the researcher and participant and lead to demand characteristics (Bargh & McKenna, 2004). It may also affect the participants willingness to withdraw from a given study or question the experimental procedures.

It is important to note that while online experiments can give greater autonomy to participants, in doing so they can also shift the locus of control away from the experimenter. The conditions in which the experiment is completed are determined by the participant, who may not be as fastidious about the experimental environment as a researcher might be. Additionally, the actual conditions in which the experiment was run are only accessible to the researcher through participant self-report.

Potential impact on the course of the field: ecological validity, new questions and issues

Finally, online experimentation can significantly impact the ecological validity of psycholinguistic research. Ecological validity (Brunswik, 1955), or the extent to which observations made rigorously controlled laboratory conditions can be generalized to more naturalistic settings has long been a core issue in the cognitive and social sciences (Harre and Secord, 1972). While online experiments do not offer a more ecologically valid methodology, they allow psycholinguists to study how language behavior might be influenced by experiment setting and potentially further ratify the validity of lab-based work. In their defense of laboratory experiments, Berkowitz and Donnerstein (1982) admit that "the meaning the laboratory participants impart to the experimental stimulation and the responses they make determines…the extent to which their laboratory reactions parallel the behavior they exhibit in other settings" (1982, p. 255). It is the mental state of the participant that matters. This will vary greatly across participants and will be determined by many factors other than setting. However, it is easy to imagine that the presence of volunteer bias, demand characteristics, power imbalance, and social pressure (Speed, Wnuk & Majid, 2017) could lead to substantial differences in participants' mental state. Online experimentation opens the possibility for setting to be manipulated as an independent variable. By comparing results of equivalent lab-based and online experiments, the degree to which experiment setting can influence language behavior can be teased out. Finding no difference across setting would support the external validity of lab-based experiments. Alternatively, a misalignment of findings would hopefully prompt a wave of welcome replication studies. This is a potential win-win situation.

What is PyschoPy3 and what can it do?

Running experiments online is not a new phenomenon. Technological developments over the past decade have been expanding the capabilities of online research through advancements in experimental software (WebExp, PsychoPy3, ScriptingRT) and the platforms (MTurk) through which they can be distributed. PsychoPy3 is the most recent open-source software for designing and creating online experiments.

PsychoPy3 (Pierce, 2019) automates the entire process of coding, uploading, and hosting experiments online. HTML and Javascript code for the experiment is generated based on the arrangement of components, routines, and loops constructed by the user in the PsychoPy3 graphic user interface. This code is then synced with Gitlab, an online repository, allowing for collaborative workflow and sharing. Finally, Pavlovia.org acts as the participant interface for the Gitlab repository, hosting the experiment on a unique URL, where participants can access the experiment. These steps are completely automated by PsychoPy3 and, once the experiment is designed, can be accomplished in under a minute with a short series of clicks.

PsychoPy3 was designed with three aims in mind: to be easy, precise, and flexible (https://www.psychopy.org/). The builder interface allows users to create a graphical representation of the experiment from which the code will be generated automatically (Peirce et al, 2019). Core components, such as auditory, visual, and text stimuli, can be added and customized, as can response measures, such as keypresses, sound capture (via microphone), and Likert scales. As we discuss in greater depth below, the presentation of stimuli and the recording of responses can be controlled with a high level of time resolution.

While on the surface, the builder interface is very simple, it is highly customizable. Each component includes a wide variety of directly customizable features, and the inclusion of a custom code component allows more experienced coders to write their own lines of code directly. Although PsychoPy was initially designed for research in psychophysics, the flexibility of the software mean that it can be easily co-opted for psycholinguistic research.

How much coding ability is required?

The intuitive, point-and-click graphic user interface and highly customizable stimulus and response components of PsychoPy3 allow for experiments to be constructed with relatively little coding knowledge (Pierce, 2007). For experiments that remain within the parameters of the available components, users will not be required to write any lines of code at all. PsychoPy automatically compiles the HTML and Javascript code necessary for the experiment to run. It is only when the requirements of the experiment exceed the limits of component functionality that coding becomes necessary. Using the custom code component in the builder, lines of Javascript can be compiled into the final script at locations specified by the users. The advantage of this feature is that it allows for increased customizability without requiring the user to fully comprehend the raw code lines of the experiment, effectively smoothing the learning curve.

An illustrative example: Lexical decision task

The requirements of a typical lexical decision experiment are met by the functionality of the PsychoPy3 builder, and thus, requires no knowledge of coding to create. Using the text stimulus component, instructions, fixation points, and stimuli can be presented to the participant, and using the keyboard response component, participants can navigate from screen to screen and indicate their lexical decision responses. Figure 3.0 shows the layout of such an experiment. A conditions file, created in excel, is loaded into the loop. During each pass through the loop, a different row of the excel file will be selected, attributing the value in that row to a variable created with the name of the header. By setting the text value of the stimuli component to the name of the header in the conditions file, the lexical stimuli will be updated with every repeat of the loop. The order in which rows are selected from the conditions file can also be set (e.g. sequential, random).

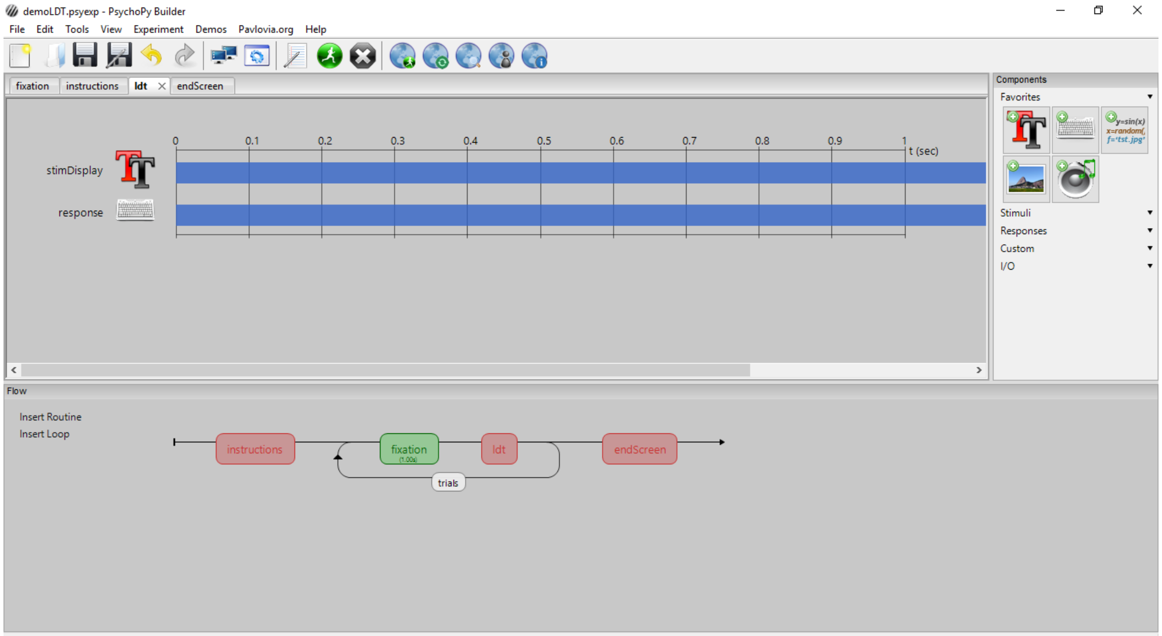

Figure 2. Example of a simple lexical decision experiment in the builder view of PsychoPy3.

Once the experiment is created in the builder, the HTML and JS code can be compiled. This is done automatically when the experiment is synced with the repository. It can also be done manually in the drop-down menu. After the code is compiled and synced to the online repository, the experiment will be available at a unique URL. For those preferring to host their experiments on their own server, the HTML and JS code compiled by PsychoPy3 can be found in the html folder created within an experiment's working directory.

Many tutorial materials exist for the purposes of explaining the basic functionality of PsychoPy . The book *Building Experiments in PsychoPy* (Pierce & MacAskill, 2018) is designed to offer both instruction and a comprehensive overview. There are web-based resources, tutorials, and demonstrations available. A selection is listed in Table 1 (since the builder view of PsychoPy3 does not differ greatly from previous versions, materials predating the release of PsychoPy3 are also useful).

*Table 1***.** Materials for learning the fundamentals of PsychoPy.

Materials for Learning PsychoPy3	
Fundamentals of PsychoPy	https://www.youtube.com/watch?v=VV6qhuQgsiI
Creating online studies from the Builder	https://www.psychopy.org/online/fromBuilder.html
Forum for technical support	https://discourse.psychopy.org/
PsychoPy3 & Pavlovia overview	https://www.youtube.com/watch?v=bprYeBzkUc8
Lexical decision and typing tasks in PsychoPy3 demonstration	http://wordsintheworld.ca/research/videos/

Matters of Technical Practicality

In this section we provide an overview of some technical matters concerning the current use of PsychoPy3. In our discussion, matters of time resolution play a large role. This is certainly understandable in the field of behavioural psycholinguistics, where research tools have typically been evaluated with respect to whether they can attain millisecond accuracy.

To better understand whether the online version of PsychoPy3 showed greater timing inconsistency than non-web-based versions, we ran tests both online through the internet browser and locally in the PsychoPy3 software. All tests were conducted on an HP laptop with 8gb of RAM, and an Intel® Core™ i5-6200U processor, a mid-range level laptop. All web-based experiments were run in the Google Chrome browser.

Stimulus timing

A limitation of all computer-based psycholinguistic research is that temporal accuracy of visual stimulus presentation is restricted by frame refresh rate. Most computer monitors have a frame refresh rate of approximately 60 frames per second. A frame rate of 60 restricts stimulus presentation to intervals of 16-17ms (~1000/60) because new visual stimuli can only be drawn to the screen at a refresh point. For example, if a stimulus is set to appear on the screen at 400ms, it should appear at the 24th frame refresh (1000/60*24=400). However, slight (sub-millisecond) variations in internal clock time or deviations in frame rate, could lead to the stimulus not appearing until in the 25th frame refresh. This would result in the largest possible delay of roughly 17ms. Stimulus presentation times nearest to refresh interval boundaries will be most susceptible to such variation. Timings nearest to the mid-point between frame refresh intervals (e.g. 392ms) should show the least variability.

A test code was developed set to draw a visual stimulus to the screen and record latency when the internal experiment clock reached 400ms (Test 1) and 392ms (Test 2). Since 392ms is the furthest point from the frame refresh interval, variability in Test 2 should be less than in Test 1. Using PsychoPy3, functionally identical test codes were created to be run locally and online. Lines of custom code were set to execute on every frame refresh. The time recorded was taken to reflect the precise time at which the visual stimulus was drawn to the screen. Descriptive statistics of the data collected from these tests is presented in Tables 2 and 3.

Table 2. Results from Test 1 (at frame refresh boundary) on stimulus presentation timing. The performance of online and offline versions of PsychoPy3 were compared. Perfect performance would result in latencies of 400 milliseconds being consistently recorded (i.e., SD =0).

	Test run locally in PsychoPy3	Test run online (Google Chrome)
Mean latency (SD) (ms)	415 (1)	414 (4)
Minimum latency (ms)	401	400
Maximum latency (ms)	417	439

Table 3. Results from Test 2 (midpoint of frame refresh) on stimulus presentation timing. The performance of online and offline versions of PsychoPy3 were compared. Perfect performance would result in latencies of 392 milliseconds being consistently recorded (i.e., SD =0).

	Test run locally in PsychoPy3	Test run online (Google Chrome)
Mean latency (SD) (ms)	398 (1)	398 (2)
Minimum latency (ms)	392	394
Maximum latency (ms)	408	427

In our test, inconsistencies in timing were greatest when they were closest to the frame refresh boundary (in this case, 400 ms). Differences between the online and offline means were small. However, we did note that the online tests showed a larger standard deviation in both cases. The maximum deviations from the expected timing were greater than 17 milliseconds and therefore cannot be solely attributed to frame refresh rate.

A further test was conducted to determine whether these larger timing deviations could have been caused by browser performance. We considered that two factors, in particular, might play an important role. They are the extent to which the browser was performing other tasks while the test was being run and the fullness of the browser cache.

We investigated these possible effects by replicating Test 1 (400 ms). We first created two conditions of cache fullness. In the 'full cache' condition, the cache contained between 300-400mb of data, while in the 'clear cache' condition, the contents of the cache were cleared immediately before running the experiment.

We also created two processing load conditions. In the 'high processing load' condition, five tabs were open in the Google Chrome browser; one for the experiment, and four others for Facebook, Reddit, the University homepage, and an hour-long music compilation playing on YouTube. This configuration was chosen to emulate a possible browser state on a university student's personal computer. In this condition, the physical memory usage of the Google Chrome browser was ~1000mb at test onset. In contrast, the low processing load condition had no other tabs open with memory usage of ~280mb at onset. Each test was run for 1000 trials. Test results are described in Table 4.

Table 4. Stimulus presentation timing under two conditions of cache fullness and two conditions of browser processing load.

Cache fullness	Full		Empty	
Processing load	Low	High	Low	High
Mean latency (SD) (ms)	413 (4)	413 (5)	413 (4)	413 (4)
Minimum latency (ms)	400	400	400	400
Maximum latency (ms)	436	443	435	435

Our test, designed to simulate maximum differences between computer states (in terms of browser cache fulness and processing load), did not yield any differences in mean latency and small differences in standard deviations. We took this as indication that web-based testing is likely robust with respect to potential variation in a participant's browser state at tome of testing. However, more testing, using a variety of computers and browsers, would be required to explore this more fully.

As a final note, the temporal accuracy of auditory stimuli is still questionable. Trial-to-trial variation is currently reported to be in the tens of milliseconds (Peirce et al, 2019), which may limit the practicality of PsychoPy3 for some online experiments using auditory stimuli. Since, PsychoPy3 relies on third-party sound libraries for processing of sound files, improvements to the temporal precision of audio stimuli cannot be accurately predicted by the Psychopy3 development team.

Response timing

Response timing was tested using a script developed using Autohotkey (https://www.autohotkey.com/), which is software that simulates keypress events with millisecond accuracy. The script generated a simulated keypress every 400 ms, initiated by pressing pre-designated hot keys: 'ctrl+s'. A test simulating and detecting keypresses was run locally and online (as in Tests 1-3 above). Each test simulated a total of 500 key presses. The results are presented in Table 5.

Table 5. Results from Test 4 (keypress response accuracy). The performance of online and offline versions of PsychoPy3 were compared. Perfect performance would result in latencies of 400 milliseconds being consistently recorded (i.e., SD =0).

	Test run locally in PsychoPy3	Test run online (Google Chrome)
Mean latency (SD) (ms)	411 (2)	413 (8)
Minimum latency (ms)	393	392
Maximum latency (ms)	429	444

Our test results were comparable to those obtained for stimulus presentation timing (Test 1, shown in Table 2). The difference between the means was small, and the test run online showed a larger standard deviation than the one run locally. As we did for the test of stimulus presentation timing, we followed this up with an investigation of the potential effects of cache fullness and processing load (with a total of 500 simulated keypresses recorded per condition). The results of this test are presented in Table 6.

Table 6. Simulated response timing under two conditions of cache fullness and two conditions of browser processing load.

Cache fullness	Full		Empty	
Processing load	Low	High	Low	High
Mean latency (SD) (ms)	411 (7)	411 (8)	412 (7)	412 (7)
Minimum latency (ms)	374	347	389	392
Maximum latency (ms)	445	490	434	444

As we have observed in previous tests, the mean latencies are very similar, as are the standard deviations. The 'full cache - high processing load' condition contained two outlier values, affecting the standard deviation and resulting in a higher maximum value (490 ms). Overall, however, we do not see that, under the simulated conditions, response time is affected by browser cache fullness or the processing loads we established. As we show below, though, it is possible to create levels of processing that will result in an experiment failing to complete.

Driving the system to failure

Experiments containing large numbers of trials can potentially push web browsers past their processing capacity. PsychoPy3 stores data collected throughout the experiment in a PsychoObject, which must be stored in the physical memory of the browser. Once the experiment is completed, the data are then converted into either a comma-separated-value (.csv) or an Excel (.xlsx) file. If the physical memory storage requirement exceeds capacity, the experiment will crash, and the data will be lost.

To determine the limits of the browser, Test 1 (stimulus presentation at 400ms) was repeated with an undefined number of trials, allowing the experiment to run until it crashed. This was repeated in high and low processing load conditions. In the high processing load condition, the browser was initially using roughly 1000mb of physical memory. After running and storing data from 1000 trials, the browser was using roughly 2439mb of memory. Including the relatively lesser requirements of other programs, the computer was using 71% of its physical memory capacity. After 2000 trials, the 3963mb was being used to run the browser (86% of capacity). At trial 2954 the experiment crashed, as memory usage hovered around 4500mb (91% capacity). In the low processing load, the memory required to run the experiment increased at rates equivalent to that observed in the high processing condition. This experiment crashed at trial 2993, once the memory requirement had reached roughly 4500mb (91% of capacity).

The implication of this finding is that the processing capacity of computers used by participants should be taken into consideration when determining the number of data points collected in an experiment. For an average laptop, the browser is only capable of holding ~3000 rows of data. This limits the types of experiments that can be administered using PsychoPy3. Similarly, limiting the number of trials for behavior experiments, such as typing tasks, is also important. Depending on the length of typing targets, the number of trials should be reduced. In the case of sentence targets with over 30 characters per sentence, the number of trials would need to be kept below 100.

Alternatively, hardware requirements can be included in selection criteria during participant recruiting.

What is new in PsychoPy3?

Embedded links

Online experiments are capable of redirecting participants to specified URLs upon completion. Demographic questionnaires and other debriefing information can be hosted on other webpages while still being part of the experiment. Participants who leave the experiment prior to completing all trials can be redirected to a page inquiring into the reasons for withdrawal and providing information regarding compensation and future participation. To deliver batteries of online tests, experiments can be linked together.

Functionality

Not all builder components are currently available for PsychoPy3. The most recently updated list of functionality is available at https://www.psychopy.org/online/status.html#onlinestatus. While certain components are available in the builder view, if they are not supported by PsychoJS (the library used to generate JS code for experiments created in the builder), they will not work when run online. If certain components are not yet available, a query can be posted to the PsychoPy forum (https://discourse.psychopy.org/) under the tab 'online experiments', which is regularly checked by the development team, to check for work-arounds or upcoming updates.

At the time of writing, the response builder components available in the PsychoJS library are keyboard, mouse, and rating scales. For stimuli image, text, sound files (wav preferred), and video files can be used. Detailed explanation of builder components are presented in Peirce (2009) and Peirce et al. (2019).

Pavlovia

Pavlovia acts as the user interface for the Gitlab repository to which PsychoPy3 experiments can be automatically synced. Experiments can optionally be made public during the syncing process. Public experiments can be accessed via searches of the Pavlovia database by other Pavlovia users or non-registered visitors to the site. For one participant to access experiments hosted on this server will require the purchase of a Pavlovia credit, priced at £0.20. Institutional licenses will also be available. Capability with Amazon's Mechanical Turk platform has recently been added to PsychoPy3, meaning that experiments hosted on Pavlovia can be embedded into MTurk windows and easily run by participants. This compatibility facilitates quick, easy, and cost effective recruitment of large numbers of participants.

The Pavlovia/Gitlab repository provides a platform for long-distance collaboration, while also creating a platform on which raw data and experimental code can be made open-access. Using the Gitlab workflow, projects can be created and updated from multiple user accounts. The contributions and changes made by each member can be tracked over time. Raw data can also be made public by adjusting the privacy settings of a repository. In this way, raw data files can be made available to anyone wishing to access them. The source code for experiments can also be viewed or duplicated (forked) through Gitlab, allowing for efficient replication of existing studies and open-access sharing of methodologies and experimental procedures. When an experiment which has been publicly shared is duplicated, a notification is sent to the creator of that repository to protect intellectual property. Access to a repository can also be granted to specific users to protect sensitive data or source code that the developer may not wish to share more broadly. Such repositories and platforms are designed to facilitate collaboration and accelerate innovation.

Discussion and Conclusions

PsychoPy3 makes psycholinguistic experimentation accessible without the resources of a fully-equipped laboratory. It expands the range of easily accessible sample populations, providing the opportunity to ask new kinds of questions about a wider range of people. It offers an intuitive interface through which online experiments can be easily constructed. However, it is relatively new software (released in its Beta version in July 2018), and is not without challenges. Regular PsychoPy3 users will likely experience some technical issues. Fortunately, PsychoPy3 provides rapid, insightful user support through the forum located on the PsychoPy website (https://discourse.psychopy.org). At the time of writing, 330 queries have been posted to the forum and 276 (83.6%) were responded to directly by the developers. Additionally, over the past year, the developers have released multiple updates, guided by user feedback, to improve user experience and rectify reported bugs and glitches. The quality of user experience with PsychoPy3 will likely continue to improve.

To mitigate several issues identified as barriers to successfully running experiments developed in PsychoPy3 via web browsers, we suggest the following three best practices.

Best practice suggestion 1: Pilot on multiple machines on multiple browsers. Our timing accuracy tests were conducted using a single browser -- Google Chrome-- and were run on a single mid-range laptop with a Windows 10 operating system. Therefore, different combinations of hardware and software could behave in ways that differ from our observations. The processing capacity of lower-end laptops may be more easily exceeded, leading to more crashes and more data loss. Thus, we suggest piloting experiments across a variety of browsers, operating systems, and computers prior to recruiting participants.

Best practice suggestion 2: Provide additional detailed instructions. Since researchers are not present to address any procedural misunderstanding or technical difficulties, optimizing instructions during the piloting stage is advisable. Instruction should also ensure that participants choose optimal conditions in which to complete the experiment. For instance, participants can be directed to find a quiet environment free of distraction, which is conducive to focus. They can be directed to connect headphones (in the case of experiments with auditory stimuli) and to turn off any background music. Although our tests show program robustness with respect to cache fullness and browser processing load, We would recommend that participants be asked to close any other tabs open in their browser and exit any non-essential program or apps running in the background.

Best practice suggestion 3: Compensate with questionnaire items. To further compensate for the lack of control over experimental conditions, self-report data from participants can be included to track differences in environment. For example, data on the level of noise or distraction in the environment can be collected. Questionnaires can be embedded into the experiment directly, using the text stimuli and keyboard response components. Additionally, a dialogue box which appears while the experiment loads, can be customized to collect participants' information. The benefit of this method is that information collected at this stage can be used to customize the filename of the data output file.

Conclusion: Is the future online?

In our view, the answer to this question is "yes" for most types of behavioural experiments not requiring specialized hardware such as ERP and eye-tracking (although the latter may turn out to be a temporary limitation).

In the history of science, revolutions within a field are often linked to technological innovations. Online experiments have already begun to change the type of data that are available to those studying psycholinguistics. The scope of data collection is enhanced through crowd-sourcing. The geographical scope of experiments has been broadened as the population of people able to access the internet increases. With introduction of intuitive experimental development software, such as PsychoPy3, the types of people who are able to conduct psycholinguistics will likely increase, bringing change to the field.

Online experiments, with the kinds of considerations and recommended adjustments discussed above, can offer a well-controlled means of investigating a much broader range of participants and situations than has so far been possible. In our view, this represents a substantial opportunity for the field and for our evolving understanding of language processing.

References

Berkowitz, L. & Donnerstein, E. (1982). External validity is more than skin deep: Some answers to criticisms of laboratory experiments. *American Psychologist*, 37(3), 245-257.

Brunswik, E. (1955). Representative design and probabilistic theory in a functional psychology. *Psychological Review*, 62,193-217.

Cattell, J. M. (1886). The time it takes to see and name objects. *Mind,* 11(41), 63–65.

Donders, F. C. (1969). On the speed of mental processes. In W. G. Koster (Ed.), *Attention and performance II. Acta psychologica* (vol. 30, pp. 412–431). (Translation of ''Over de snelheid van psychische processen. Onderzoekingen gedaan in het Physiologisch Laboratorium der Utrechtsche Hoogeschool, 1868–1869, Tweede reeks, II, 92–120'').

Harre, R., & Secord, P. F. (1972). *The explanation of social behaviour.* Oxford, England: Blackwell,

Meyer, D. E., & Schvaneveldt, R. W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 90, 227–234.

Peirce, J. W. (2007). PsychoPy—Psychophysics software in Python. *Journal of Neuroscience Methods*, 162, 8–13.

Peirce, J. W., & MacAskill, M. R. (2018). *Building Experiments in PsychoPy.* London: Sage.

Peirce, J. W., Gray, J. R., Simpson, S., MacAskill, M. R., Höchenberger, R., Sogo, H., Kastman, E., Lindeløv, J. (2019). PsychoPy2: experiments in behavior made easy. *Behavior Research Methods.*

Peirce, J.W. (2009). Generating stimuli for neuroscience using PsychoPy. *Frontiers in Neuroinformatics*, 2, 10.

Speed, L., Wnuk, E., & Majid, A. (2017). Studying psycholinguistics outside the lab. In A. M. B. de Groot and P. Hoogart (eds), *Research Methods in Psycholinguistics and the Neurobiology of Language : A Practical Guide* (pp.190-207). John Wiley & Sons, Incorporated.