Online Experiments for Language Scientists

Kenny Smith

kenny.smith@ed.ac.uk

This is mainly a pointer to other resources

These slides and code are available online for more leisurely perusal

- https://github.com/kennysmithed/oels_emlar2024

Lots of more detailed materials on my Edinburgh undergrad/Masters course page!

- 2023 iteration: https://kennysmithed.github.io/oels2023/
- Longer lecture slides, papers, detailed notes on crowdsourcing
- Example code for linguistics experiments of various complexities
- Linguistics undergrad/MSc students with little prior coding experience all produce a working experiment by the end of the course

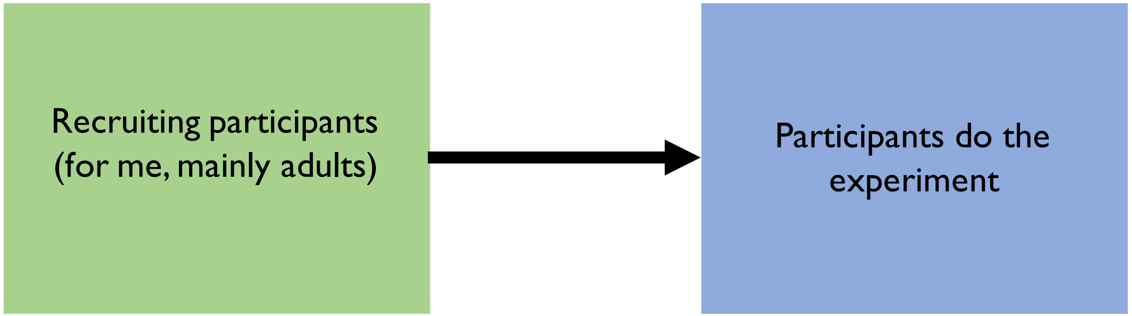
Why run experiments online?

- Pandemics!
- **Faster**
- **Larger samples**
- Potentially more diverse populations
- Access to specific linguistic populations

Challenges of doing online data collection

- It’s new (for some of us)
- Involves more components to take care of
- Involves (more) coding
- Less control:
  - Who are the participants?
  - In what environment do they participate in the study?

Running experiments



Running experiments in the lab

Various ways to recruit participants to come into the lab (SONA, advertising on campus…)

Programming environment to create the experiment (PsychoPy, Eprime…)

Supporting files & data are saved on the lab's computer

Running experiments online

Participants are recruited to do experiments on their **own** devices

Programming environment to create the experiment (so it is presented on participants' devices)

Supporting files & data are saved outside of participants' devices

Running experiments online

Platforms to recruit and compensate participants (Prolific, Mechanical Turk…)

Programming environment to create the experiment (javascript, jsPsych, Gorilla…)

Server (university or private server, Pavlovia, cognition.run, …)

Overview of this tutorial

Platforms to recruit and compensate participants (**Prolific**, Mechanical Turk…)

Programming environment to create the experiment (**javascript**, **jsPsych**, Gorilla…)

Server (university or private server, Pavlovia, cognition.run, …)

Recruiting participants online

Crowdsourcing

Once you have an experiment that runs in a browser, you can get participants from anywhere, including crowdsourcing sites

- Websites with populations of “workers” who will do online tasks for money

MTurk and Prolific

Amazon Mechanical Turk

https://www.mturk.com

- Designed for crowdsourcing **anything**
- Very light touch
- More US-based participants?
- Interface is pretty horrible (particularly for experimenter) but has a powerful API for code-based payment etc
- More chaotic, worse data (or more need to restrict participation to established workers)?

Prolific (formerly “Prolific Academic”)

https://www.prolific.co

- Designed for scientific data collection
- Heavier vetting of participants
- More UK/EU participants?
- Nicer web interface, recently added an API (but I haven’t used it)
- Maybe better-behaved participants

A look around Prolific

- From a participant perspective
- From an experimenter perspective

Recruiting children for online studies

- We (actually my colleague Shira Tal) do this
  - Older kids (age 5-6) recruited through our participant pool
  - Zoom call with screen sharing and an experiment coded in jsPsych
  - Quick compared to lab data collection
- There are crowdsourcing sites for child participants
  - https://childrenhelpingscience.com (now merged with / “powered by” lookit - mainly preferential looking, using their own jsPsych-like framework)

Pros and cons of crowdsourcing experimental data

Pros

- Not face-to-face
- Large samples, fast
- Access different populations
- \+ for replicability

Cons

- Expensive (**not** cheap)
- Lack of control
- Encourages dumb experiments?
- \- for replicability

Pro: not face-to-face

Pro: large samples, fast

MTurk and Prolific both have large active populations of workers/participants (100,000s of registered people)

- Although not everyone is active all the time
- Estimating Mturk population size is complicated (see e.g. Difallah et al., 2018)
- Prolific gives you an estimate of available and active population size

In practice, you can recruit **100s/1000s of participants in days**.

Pro: access different populations

Typical lab-based studies will sample from university student population

- Mostly undergraduates
- Mostly young
- All highly educated
- In Edinburgh, mainly native English speakers (obviously varies between unis)

If you want to access a different population, crowdsourcing might let you do that

From Pavlick et al. (2014)

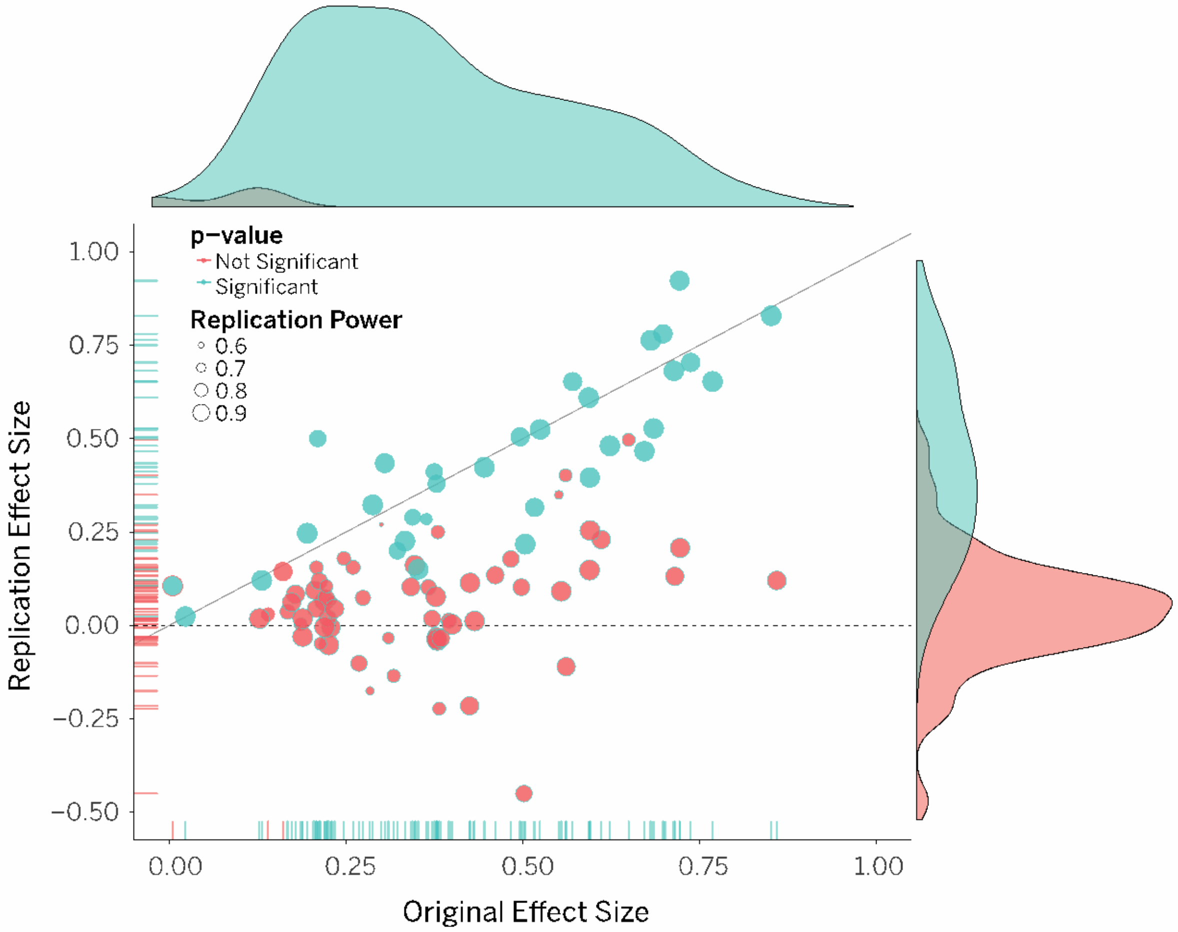
workers	quality	speed	
many	high	fast	Dutch, French, German, Gujarati, Italian, Kannada, Malayalam, Portuguese, Romanian, Serbian, Spanish, Tagalog, Telugu
		slow	Arabic, Hebrew, Irish, Punjabi, Swedish, Turkish
	low or medium	fast	Hindi, Marathi, Tamil, Urdu
		slow	Bengali, Bishnupriya Manipuri, Cebuano, Chinese, Nepali, Newar, Polish, Russian, Sindhi, Tibetan
few	high	fast	Bosnia, Croatian, Macedonian, Malay, Serbo-Croatian
		slow	Afrikaans, Albanian, Aragonese, Asturian, Basque, Belarusian, Bulgarian, Central Bicolano, Czech, Danish, Finnish, Galacian, Greek, Haitian, Hungarian, Icelandic, Ilokano, Indonesian, Japanese, Javanese, Kapampangan, Kazakh, Korean, Lithuanian, Low Saxon, Malagasy, Norwegian (Bokmal), Sicilian, Slovak, Slovenian, Thai, UKranian, Uzbek, Waray-Waray, West Frisian, Yoruba
	low or medium	fast	–
		slow	Amharic, Armenian, Azerbaijani, Breton, Catalan, Georgian, Latvian, Luxembourgish, Neapolitian, Norwegian (Nynorsk), Pashto, Piedmontese, Somali, Sudanese, Swahili, Tatar, Vietnamese, Walloon, Welsh
none	low or medium	slow	Esperanto, Ido, Kurdish, Persian, Quechua, Wolof, Zazaki

Pro: + for replicability

If you see a result in a scientific paper, can you assume that the effect they report is real and not just, e.g., a statistical fluke?

One way to check: replication

- Take someone else's experiment, replicate it, check you get the same result



From Open Science Collaboration, 2015

Pro: + for replicability

If you see a result in a scientific paper, can you assume that the effect they report is real and not just, e.g., a statistical fluke?

One way to check: replication

- Take someone else's experiment, replicate it, check you get the same result

Multiple potential advantages for online data collection

- Because collecting a large sample is easy, small-sample experiments (which are more prone to statistical flukes) can be avoided
- Because collecting data online is fast and easy, it makes it easier to replicate experiments (including your own!)
- Because populations are shared, makes it easy to replicate closely (avoiding e.g. "ah it's because your population is different" responses to non-replication)

Con: expensive (not cheap)

Mturk does not set minimum pay rates

Prolific does, but they are low (£6/hour)

These rates are for the National Living Wage (for those aged 23 and over) and the National Minimum Wage (for those of at least school leaving age). The rates change on 1 April every year.

	23 and over	21 to 22	18 to 20	Under 18	Apprentice
April 2023	£10.42	£10.18	£7.49	£5.28	£5.28

https://www.gov.uk/national-minimum-wage-rates

Con: expensive (**not cheap**)

Mturk does not set minimum pay rates

Prolific does, but they are low (£6/hour)

But we should not be paying at those rates

- It’s unethical
- It’s exploitative

Additionally

- Mturk and Prolific charge fees: 20-40% on top of what goes to participant
- Plus sample sizes tend to be bigger (because data quality can be lower or just because you can)

Con: Lack of control

In a normal lab study

- You interact with your participants when they arrive, and can see that they are indeed e.g. a sober human who speaks English natively
- They take part in a quiet, controlled lab environment on a modern machine that behaves in a known way
- You can monitor them as they participate, and they know this

With crowdsourced participants participating remotely, none of these things are true

- Consequently, experiments need to be designed to handle this

Some ways to compensate for lack of control

- Add checks on who the participants are: native language checks, instruction comprehension checks, …
- Add attention checks during the task, identify (and eject?) people who are not attending or who are responding randomly
- Can you make it easier to pay attention than not?
- Make the experiment short and fun! Most tasks on these platforms are pretty dull.

Con: encourages dumb experiments (?)

No hard constraints, but because of the lack of control, stuff that works best involves constrained and low-effort responses

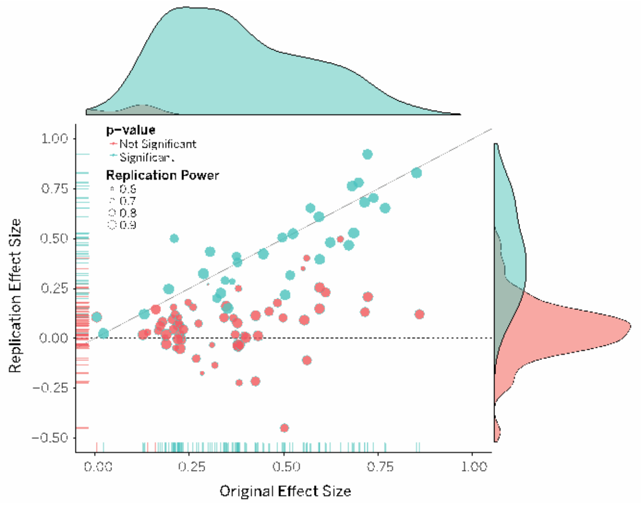
- One-off decisions (i.e. not involving complex integration of info)
- Few restricted choices per trial (not e.g. open-ended typing)
- Short experiments (a few minutes rather than an hour)

Can you investigate the questions you want using these sorts of methods?

Con: - for replicability

If you see a result in a scientific paper, can you assume that the effect they report is real and not just, e.g., a statistical fluke?

Potential risk of online data collection: Because collecting data online is fast and easy, it makes it easier to run lots of experiments, and just publish the ones that "work" (cf. "the file drawer problem")



Final note: Comparability with lab data

People often want to know if crowdsourced data is like lab data (i.e. do effects shown in the lab replicate online?)

- Lab data as a “gold standard” due to higher levels of control
- Or just because the effect you are interested in has only been shown in the lab

Effects of many paradigms first tested in the lab were replicated with crowdsourced populations (e.g., Monroe et al., 2010; Sprouse, 2011; Enochson & Culbertson, 2015; Loy & Smith, 2021; …)

Some other quick remarks on your behaviour on crowdsourcing sites

- Pay fairly
- Test and test before putting your study up
  - If things break or take a lot longer than advertised, participants get stressed
- Reject infrequently, if at all
- Respond promptly and politely to questions, don’t get into arguments
- **If in doubt, pay out**

Questions?

Building online experiments using javascript and jsPsych

Javascript and jsPsych

Javascript: a programming language that runs in web browsers

jsPsych: a library that makes it easy to build experiments (https://www.jspsych.org)

de Leeuw, J. R. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a web browser. *Behavior Research Methods*, *47*, 1-12. doi:10.3758/s13428-014-0458-y.

Josh de Leeuw
Vassar College

Resources for teaching yourself jsPsych

- My course: https://kennysmithed.github.io/oels2023/
- The main jsPsych webpage: https://www.jspsych.org
- jsPsych discussion forum on github: https://github.com/jspsych/jsPsych/discussions
- Plus for general javascript help, googling until you find an answer you can use on StackOverflow!


2

Plugins and timelines

Plugins: basic building blocks

```
var hello_trial = {
    type: jsPsychHtmlKeyboardResponse,
    stimulus: 'Hello world!'
  }
```

Timeline: a sequence of those building blocks

```
var timeline = [hello_trial];
```

A wide range of plugins available

See https://www.jspsych.org/7.3/plugins/list-of-plugins/

Building an experiment involves

- Knowing how to use plugins
- Figuring out how to piece them together
- Some tiny bits of html and javascript to connect the plugins and make them do what you want
- (Occasionally, and optionally, making your own plugin)
- **Massive reuse of code within and across experiments!**

Common components of experiments I build (usually artificial language learning tasks)

1. Consent and instructions
   - Minimally, some text + a button click
2. Training
   - Stimuli can be images, text, sounds
   - Entirely passive, or **responses required to maintain engagement**?
3. Test
   - Stimuli can be images, text, sounds
   - Responses required: button clicks, key presses, spoken responses
4. Demographics/additional info
   - Radio buttons, drop-down menus, numbers, free-text comments, …

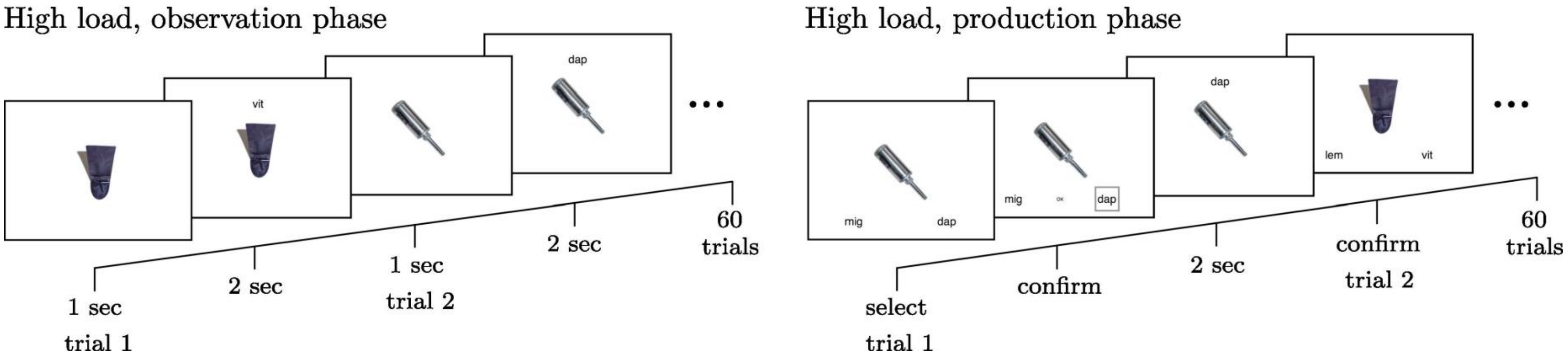
A simple frequency-learning experiment

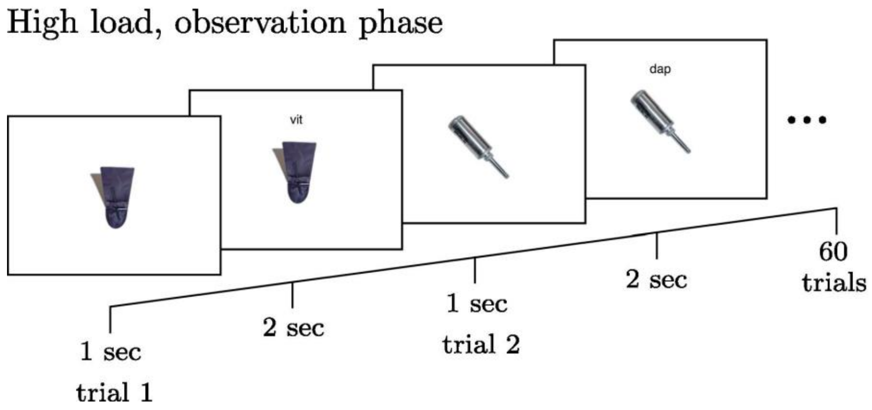
Ferdinand, V., Kirby, S., & Smith, K. (2019). The cognitive roots of regularization in language. *Cognition, 184,* 53-68.

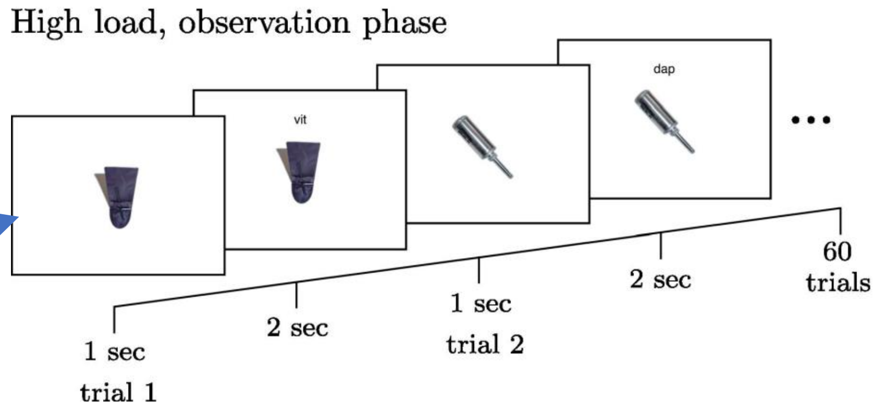
Under what conditions do participants accurately track frequency distributions in their input, and when do they *regularize* (over-use frequent variants)?

- Standard paradigm in the artificial language learning literature, e.g. Hudson Kam & Newport, 2005; Vouloumanos, 2008; Reali & Griffiths, 2009; Smith & Wonnacott, 2010; Culbertson et al., 2012; …

We were interested in effects of domain (linguistic vs non-linguistic stimuli) and demand (number of frequency distributions you have to track)

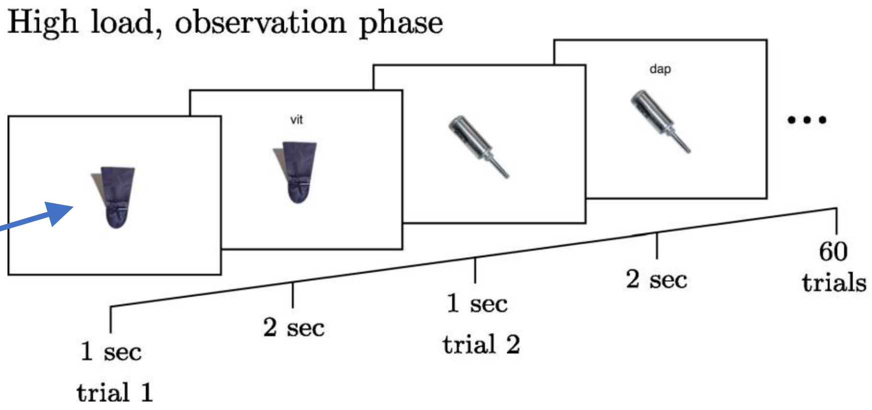

High load, observation phase
vit
dap
1 sec
trial 1
2 sec
1 sec
trial 2
2 sec
60
trials
High load, production phase
mig
dap
mig
dap
dap
lem
vit
select
trial 1
confirm
2 sec
confirm
trial 2
60
trials


High load, observation phase
vit
dap
1 sec
trial 1
2 sec
1 sec
trial 2
2 sec
60
trials



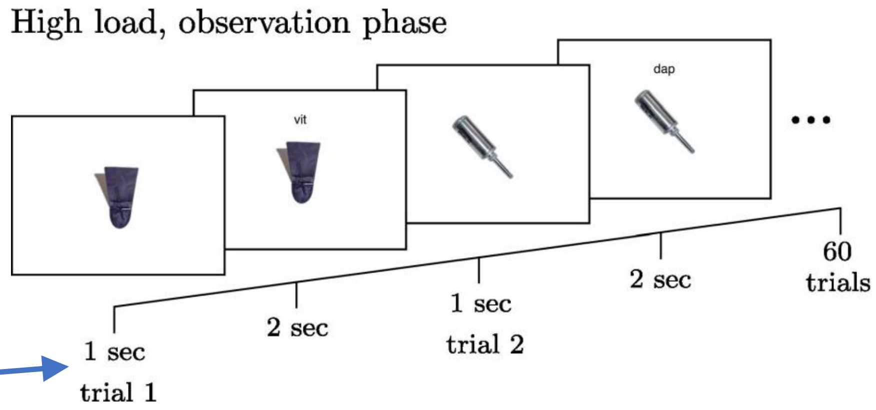
Using an image-button-response plugin to show image

```
var object2_show = {type:jsPsychImageButtonResponse,
                    stimulus:'object2.jpg',
                    trial_duration:1000,
                    choices:[]}
```



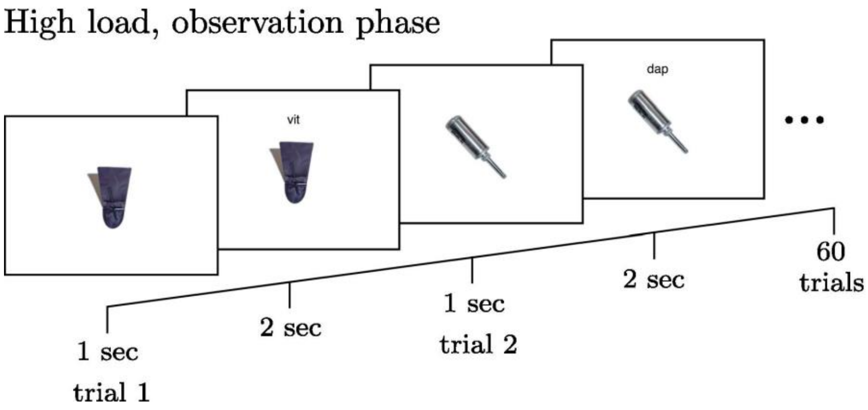
This specific image is called object2.png

```
var object2_show = {type:jsPsychImageButtonResponse,
                    stimulus:'object2.jpg',
                    trial_duration:1000,
                    choices:[]}
```



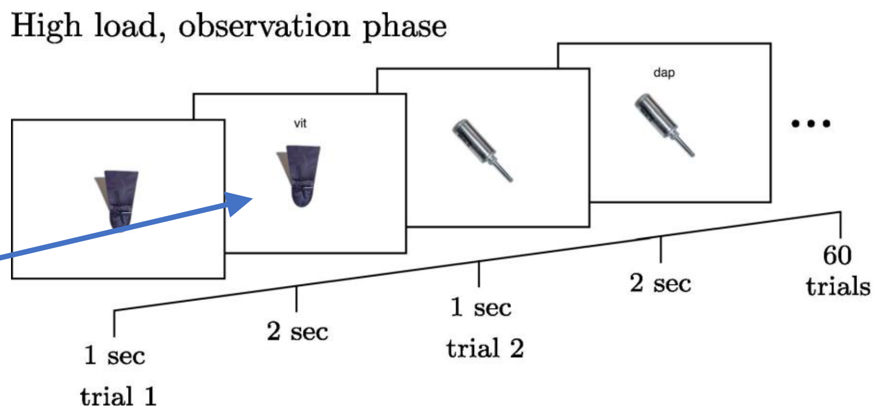
Show it for 1000ms

```
var object2_show = {type:jsPsychImageButtonResponse,
                    stimulus:'object2.jpg',
                    trial_duration:1000,
                    choices:[]}
```



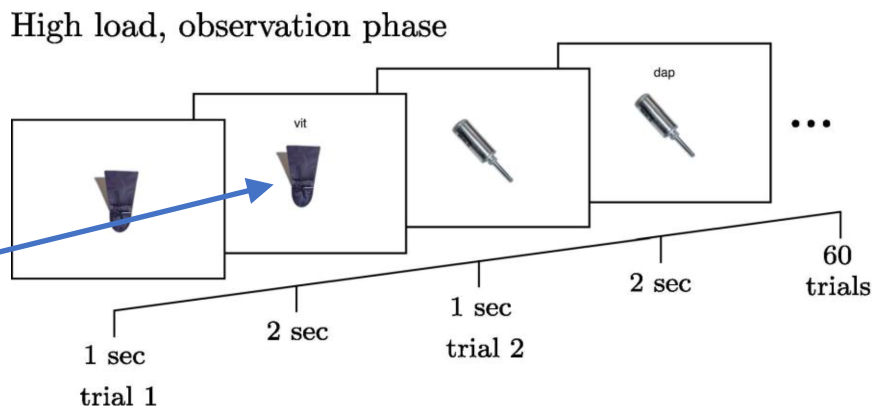
No response required!

```
var object2_show = {type:jsPsychImageButtonResponse,
                    stimulus:'object2.jpg',
                    trial_duration:1000,
                    choices:[]}
```



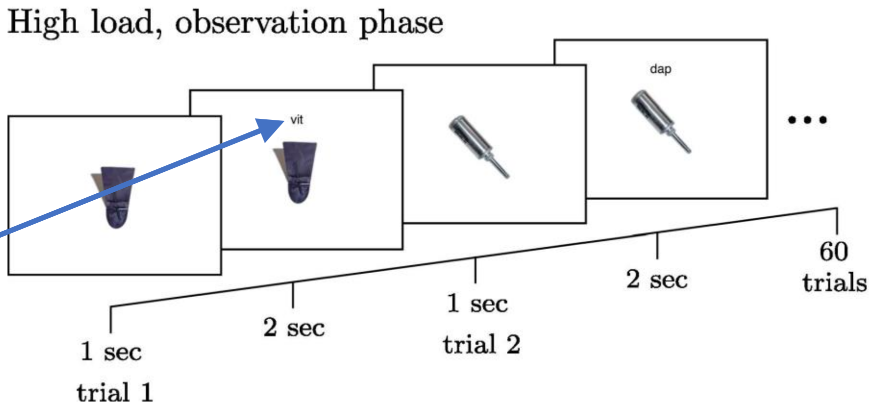
Using an image-button-response plugin to show image

```
var object2_train = {type:jsPsychImageButtonResponse,
                     stimulus:'object2.jpg',
                     prompt: 'vit',
                     trial_duration:2000,
                     choices:[]}
```



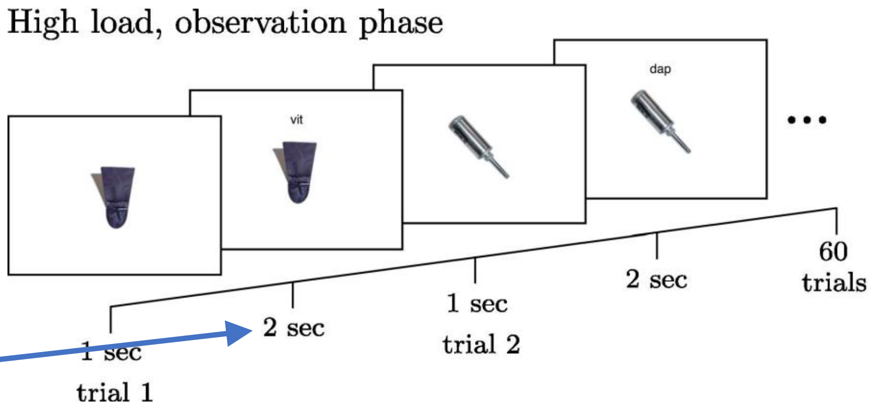
This specific image is called object2.png

```
var object2_train = {type:jsPsychImageButtonResponse,
                     stimulus:'object2.jpg',
                     prompt: 'vit',
                     trial_duration:2000,
                     choices:[]}
```



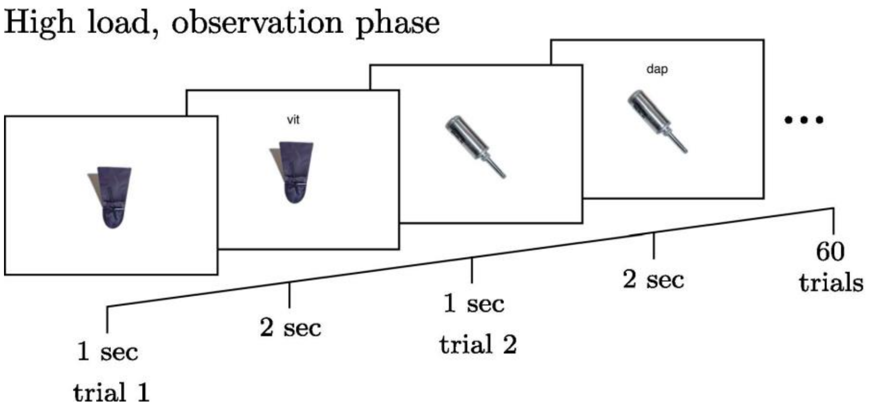
Show it along with label using the **prompt** parameter

```
var object2_train = {type:jsPsychImageButtonResponse,
                     stimulus:'object2.jpg',
                     prompt: 'vit',
                     trial_duration:2000,
                     choices:[]}
```



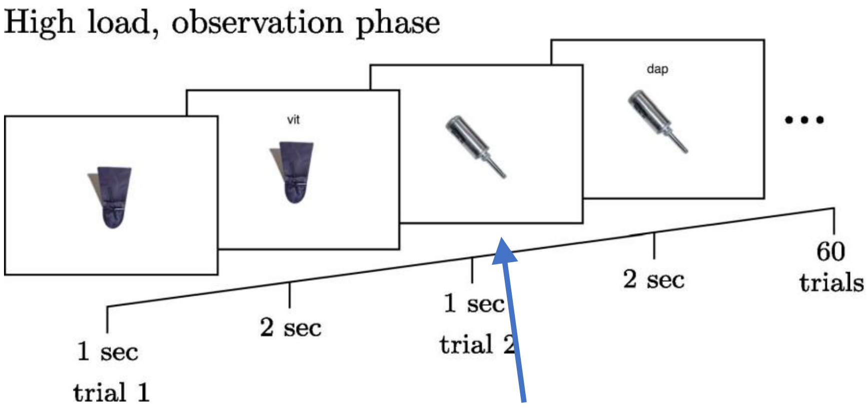
Show it for 2000ms

```
var object2_train = {type:jsPsychImageButtonResponse,
                      stimulus:'object2.jpg',
                      prompt: 'vit',
                      trial_duration:2000,
                      choices:[]}
```



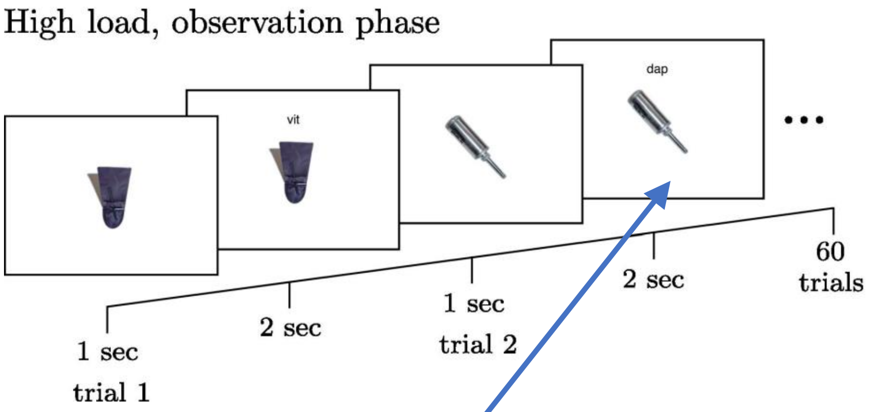
No response required!

```
var object2_train = {type:jsPsychImageButtonResponse,
                     stimulus:'object2.jpg',
                     prompt: 'vit',
                     trial_duration:2000,
                     choices:[]}
```



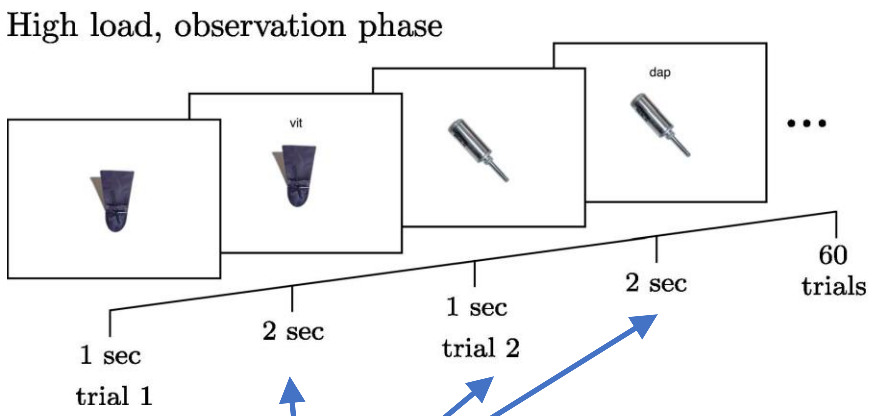
Show (different) image

```
var object4_show = {type:jsPsychImageButtonResponse,
                    stimulus:'object4.jpg',
                    trial_duration:1000,
                    choices:[]}
```



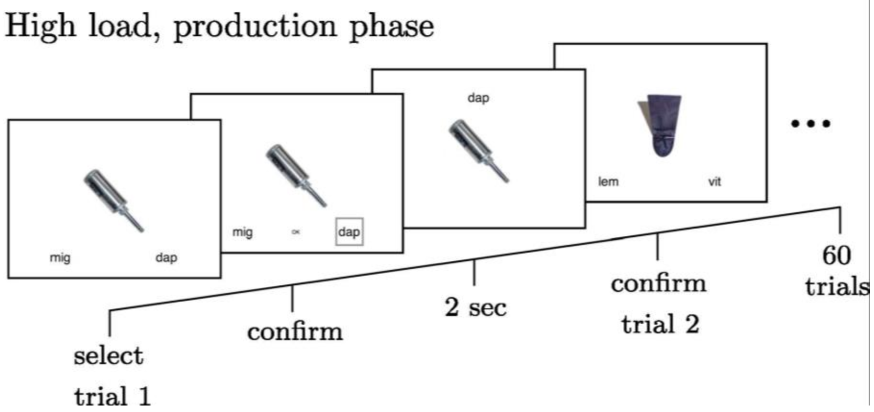
Show (different) image + label

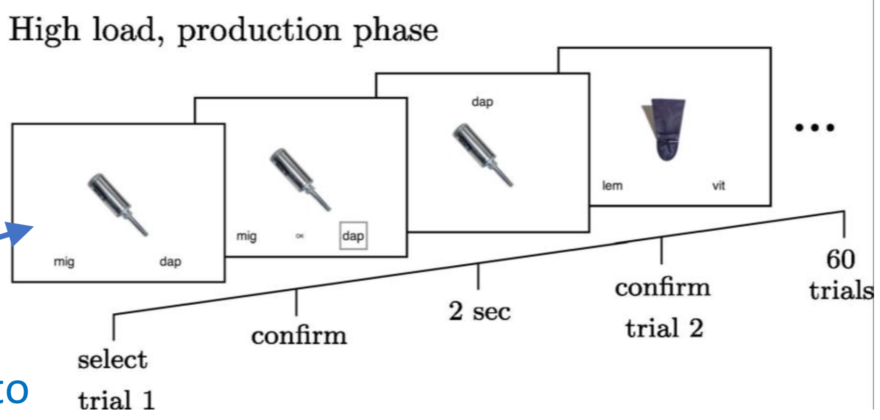
```
var object4_train = {type:jsPsychImageButtonResponse,
                     stimulus:'object4.jpg',
                     prompt: 'dap',
                     trial_duration:2000,
                     choices:[]}
```



String many trials together to form a **timeline**

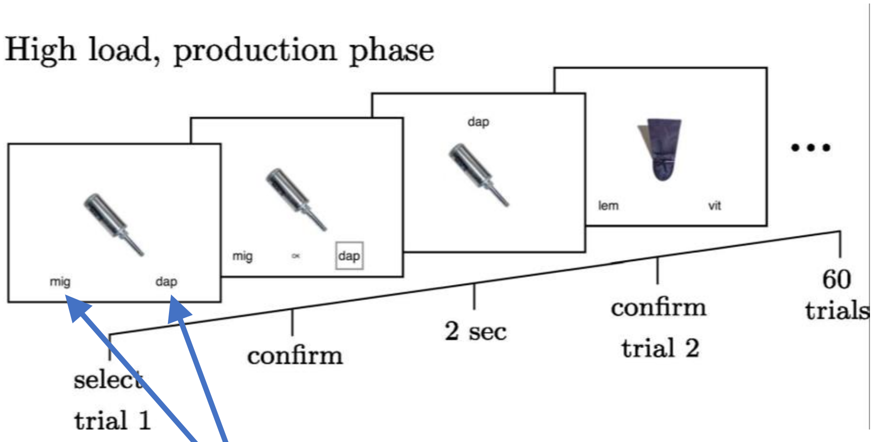
```
var training_timeline = [object2_show,object2_train,object4_show,object4_train]
```


High load, production phase
mig
dap
mig
ok
dap
dap
lem
vit
select
trial 1
confirm
2 sec
confirm
trial 2
60
trials



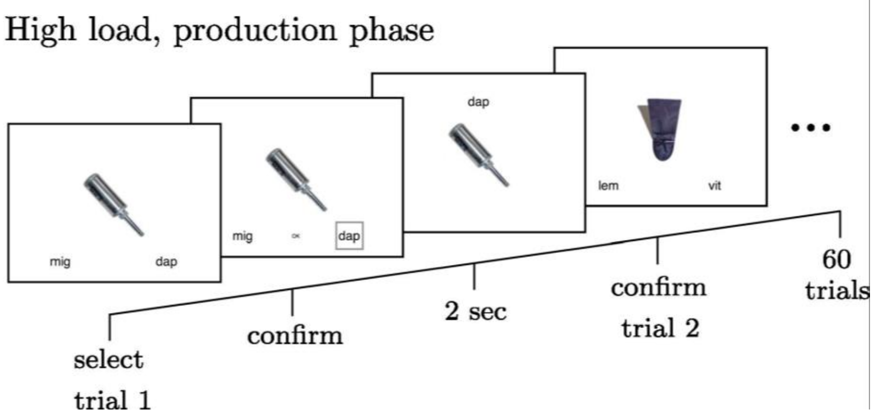
Using an image-button-response plugin to show image **and get response**

```
var object4_test = {type:jsPsychImageButtonResponse,
                    stimulus:'object4.jpg',
                    choices: ['mig','dap']}
```



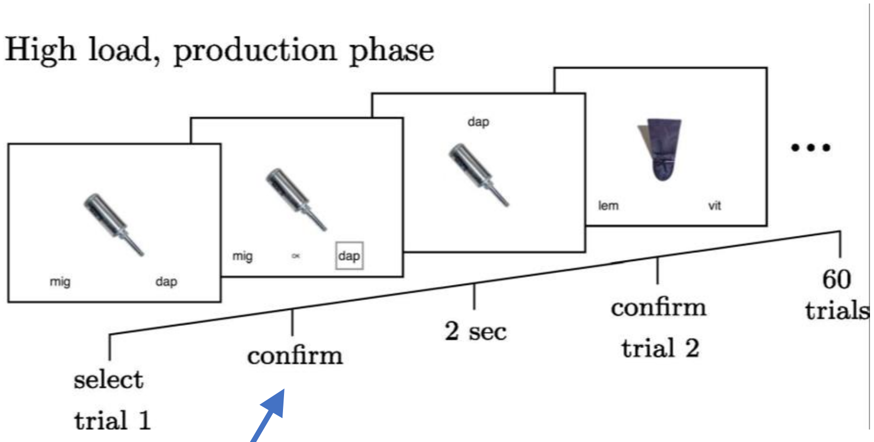
choices parameter contains labels for buttons

```
var object4_test = {type:jsPsychImageButtonResponse,
                    stimulus:'object4.jpg',
                    choices: ['mig','dap']}
```

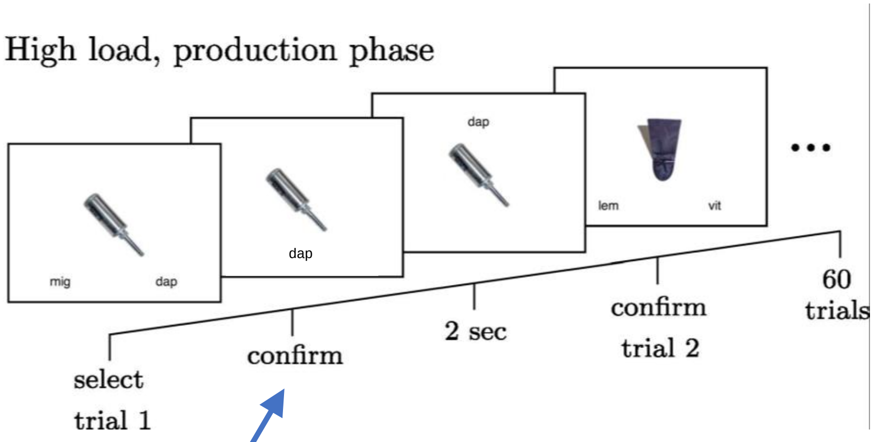


Note we no longer specify the trial duration: trial lasts until the participant responds

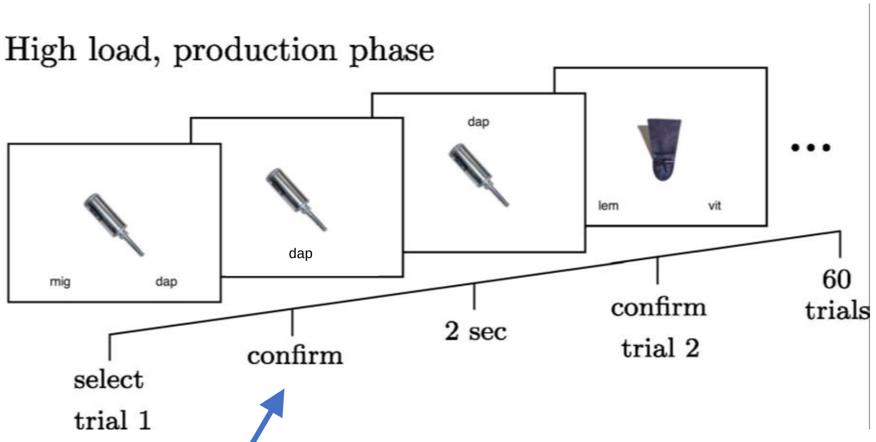
```
var object4_test = {type:jsPsychImageButtonResponse,
                    stimulus:'object4.jpg',
                    choices: ['mig','dap']}
```



Confirm trial: going to do this slightly differently and make them click on a button showing the label they just selected

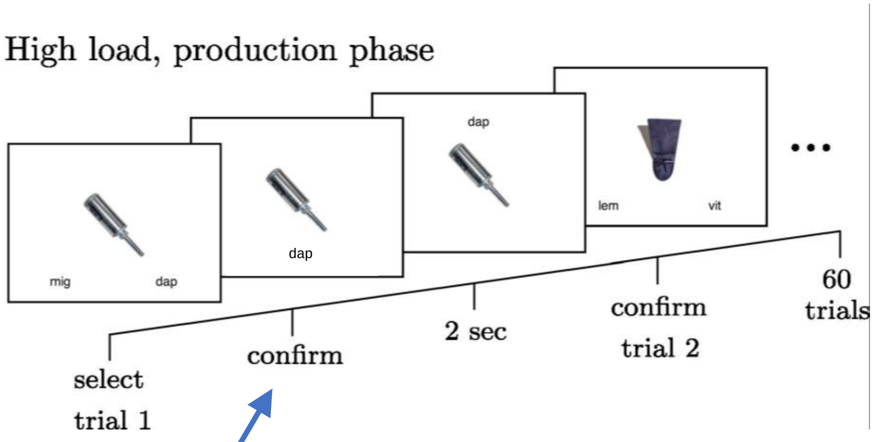


Confirm trial: going to do this slightly differently and make them click on a button showing the label they just selected



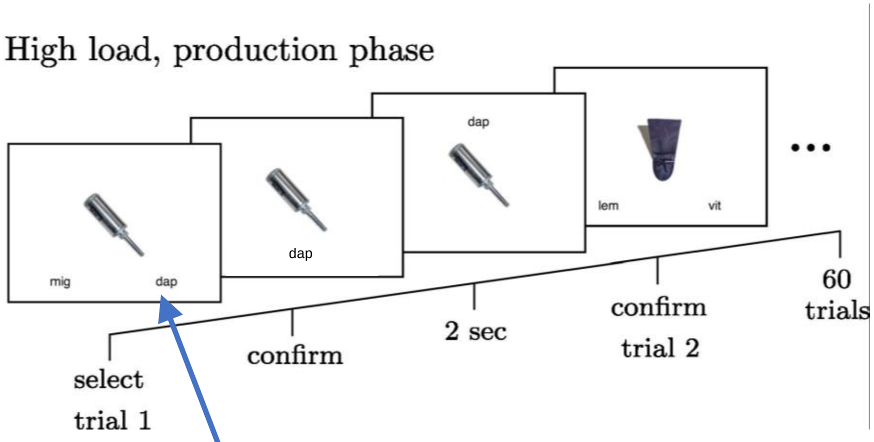
When we build the timeline (i.e. when the experiment is loaded), we don't know what the confirmation label will be!

```
var object4_confirm = {type:jsPsychImageButtonResponse,
                      stimulus:'object4.jpg',
                      choices: [],
                      on_start:function(trial) {
                       var last_trial_label = jsPsych.data.get().last(1).values()[0].label_selected
                       trial.choices=[last_trial_label]}}
```



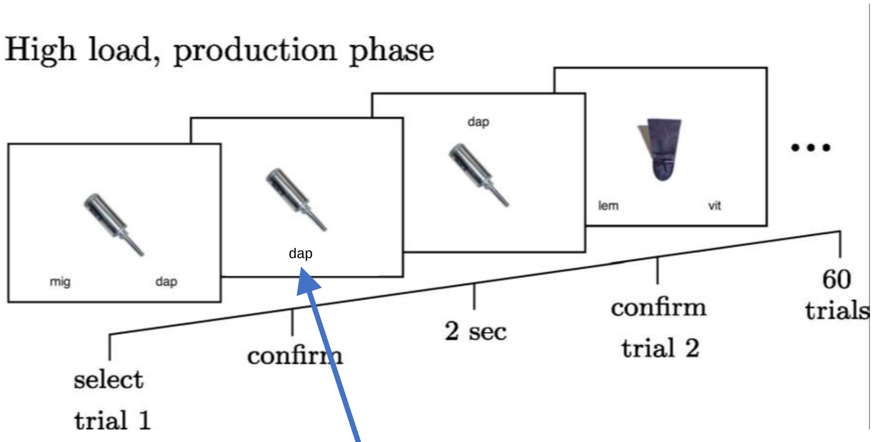
But by the time this trial starts, we can retrieve it…

```
var object4_confirm = {type:jsPsychImageButtonResponse,
                       stimulus:'object4.jpg',
                       choices: [],
                       on_start:function(trial) {
                        var last_trial_label = jsPsych.data.get().last(1).values()[0].label_selected
                        trial.choices=[last_trial_label]}}
```



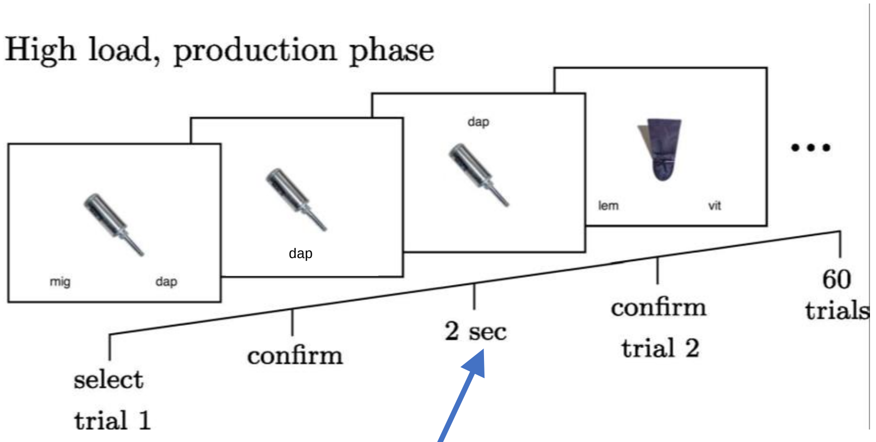
...by consulting the data generated by the immediately preceding trial

```
var object4_confirm = {type:jsPsychImageButtonResponse,
                       stimulus:'object4.jpg',
                       choices: [],
                       on_start:function(trial) {
                        var last_trial_label = jsPsych.data.get().last(1).values()[0].label_selected
                        trial.choices=[last_trial_label]}}
```



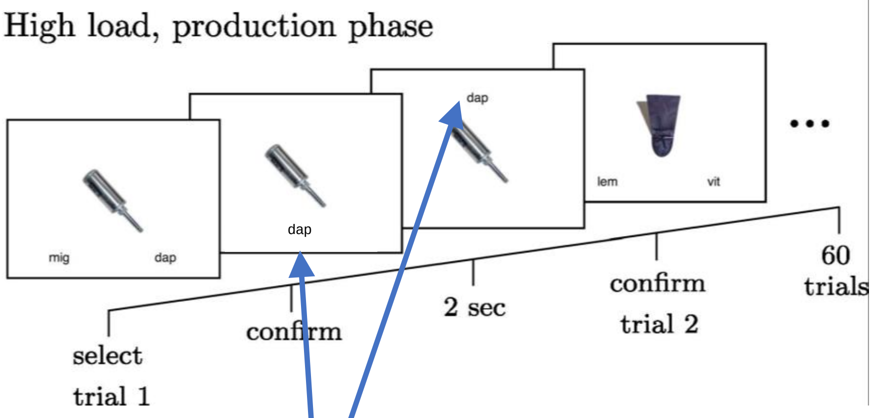
That will be the label for our confirmation button

```
var object4_confirm = {type:jsPsychImageButtonResponse,
                       stimulus:'object4.jpg',
                       choices: [],
                       on_start:function(trial) {
                        var last_trial_label = jsPsych.data.get().last(1).values()[0].label_selected
                        trial.choices=[last_trial_label]}}
```



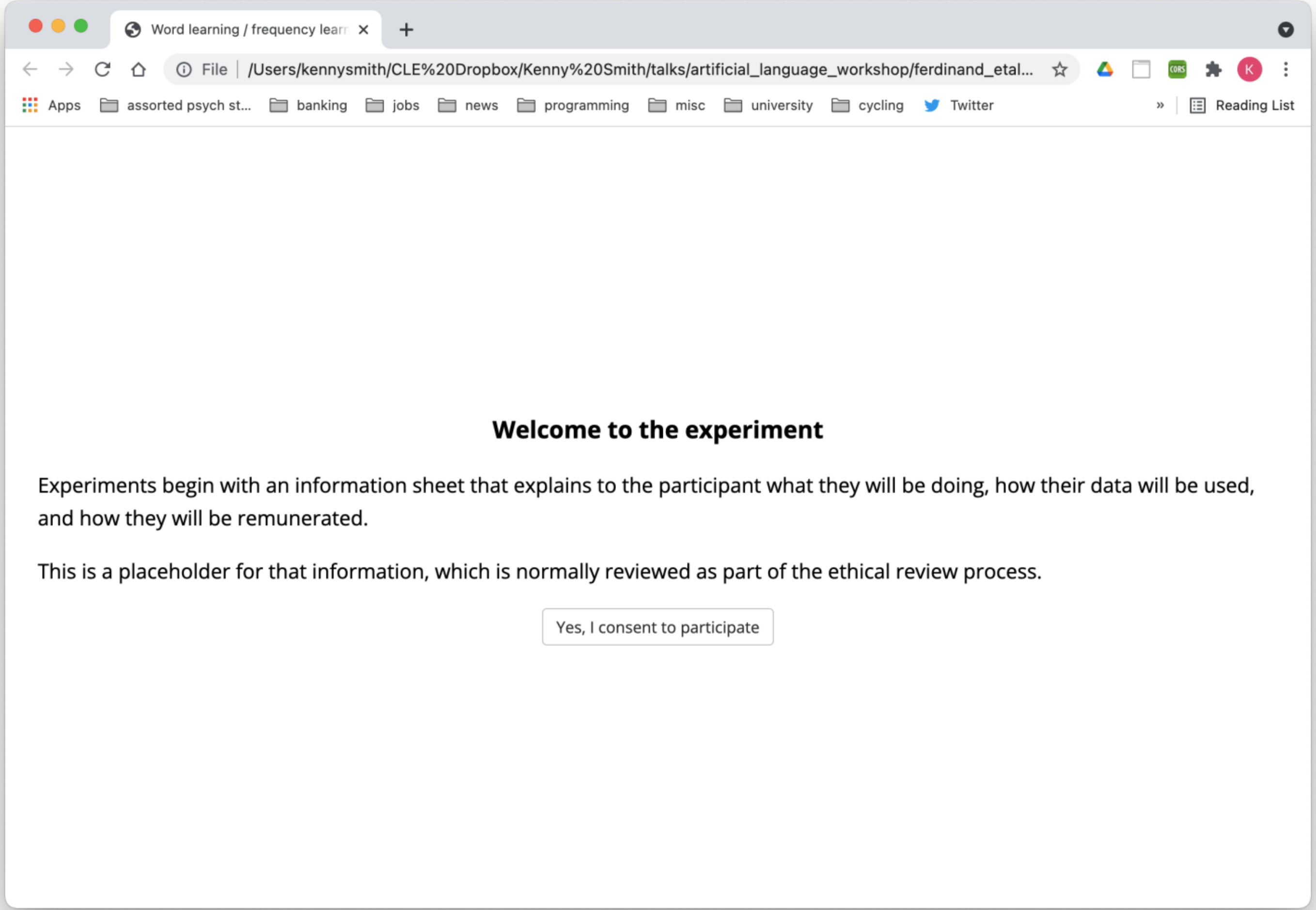
Post-confirmation display looks like a training trial…

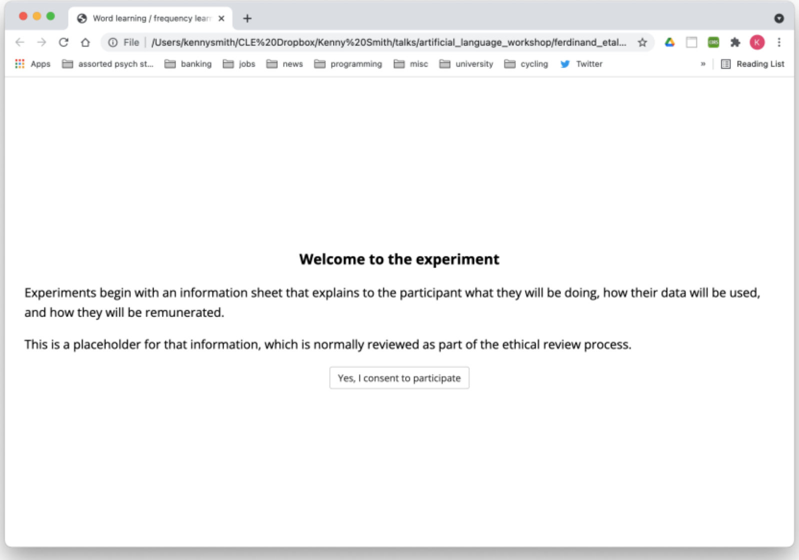
```
var object4_display = {type:jsPsychImageButtonResponse,
                       stimulus:'object4.jpg',
                       trial_duration:2000,
                       choices:[],
                       on_start:function(trial) {
                        var last_trial_label = jsPsych.data.get().last(1).values()[0].label_selected
                        trial.prompt=last_trial_label}}
```



...except we use the same trick to get the prompt from their previous trial response

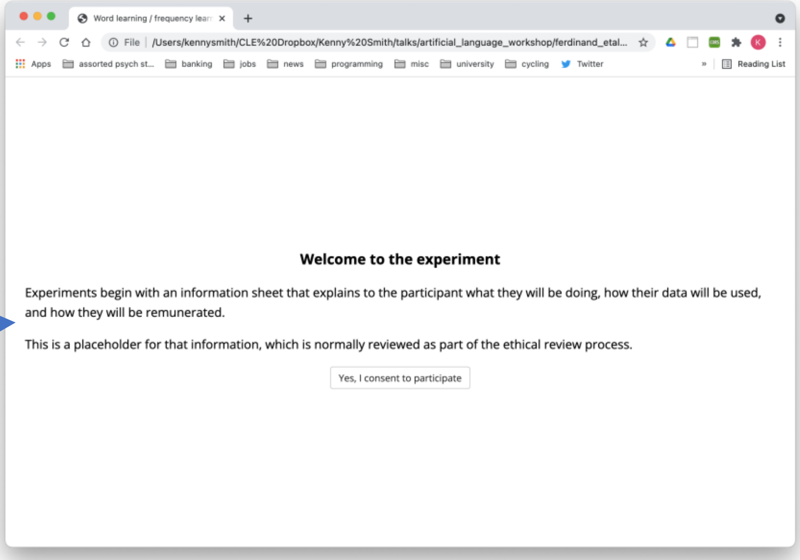
```
var object4_display = {type:jsPsychImageButtonResponse,
                       stimulus:'object4.jpg',
                       trial_duration:2000,
                       choices:[],
                       on_start:function(trial) {
                        var last_trial_label = jsPsych.data.get().last(1).values()[0].label_selected
                        trial.prompt=last_trial_label}}
```


Word learning / frequency lear
File
/Users/kennysmith/CLE%20Dropbox/Kenny%20Smith/talks/artificial_language_workshop/ferdinand_etal...
Apps
assorted psych st...
banking
jobs
news
programming
misc
university
cycling
Twitter
Reading List
Welcome to the experiment
Experiments begin with an information sheet that explains to the participant what they will be doing, how their data will be used, and how they will be remunerated.
This is a placeholder for that information, which is normally reviewed as part of the ethical review process.
Yes, I consent to participate



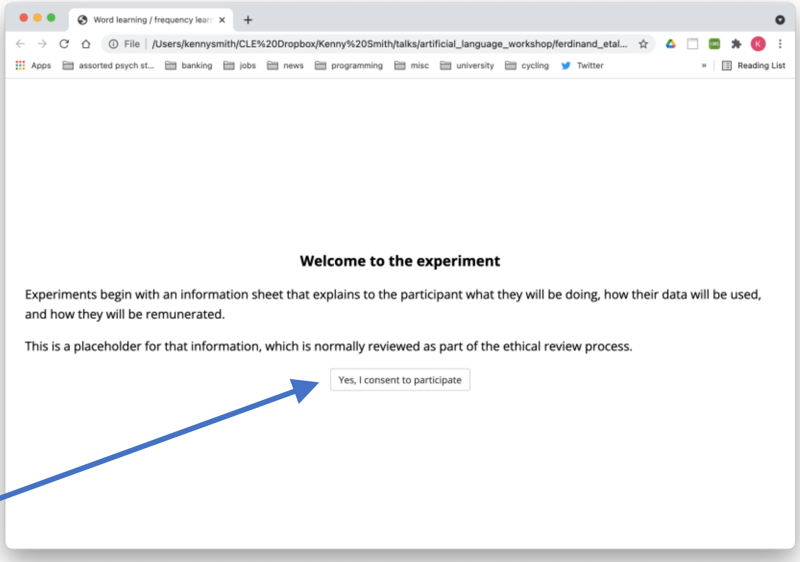
Using an html-button-response plugin to show formatted text and get them to click

```
var consent_screen = {
  type: jsPsychHtmlButtonResponse,
  stimulus:
    "<h3>Welcome to the experiment</h3> \
  <p style='text-align:left'>Experiments begin with an information sheet that explains to the participant \
  what they will be doing, how their data will be used, and how they will be \
  remunerated.</p> \
  <p style='text-align:left'>This is a placeholder for that information, which is normally reviewed \
  as part of the ethical review process.</p>",
  choices: ["Yes, I consent to participate"],
};
```



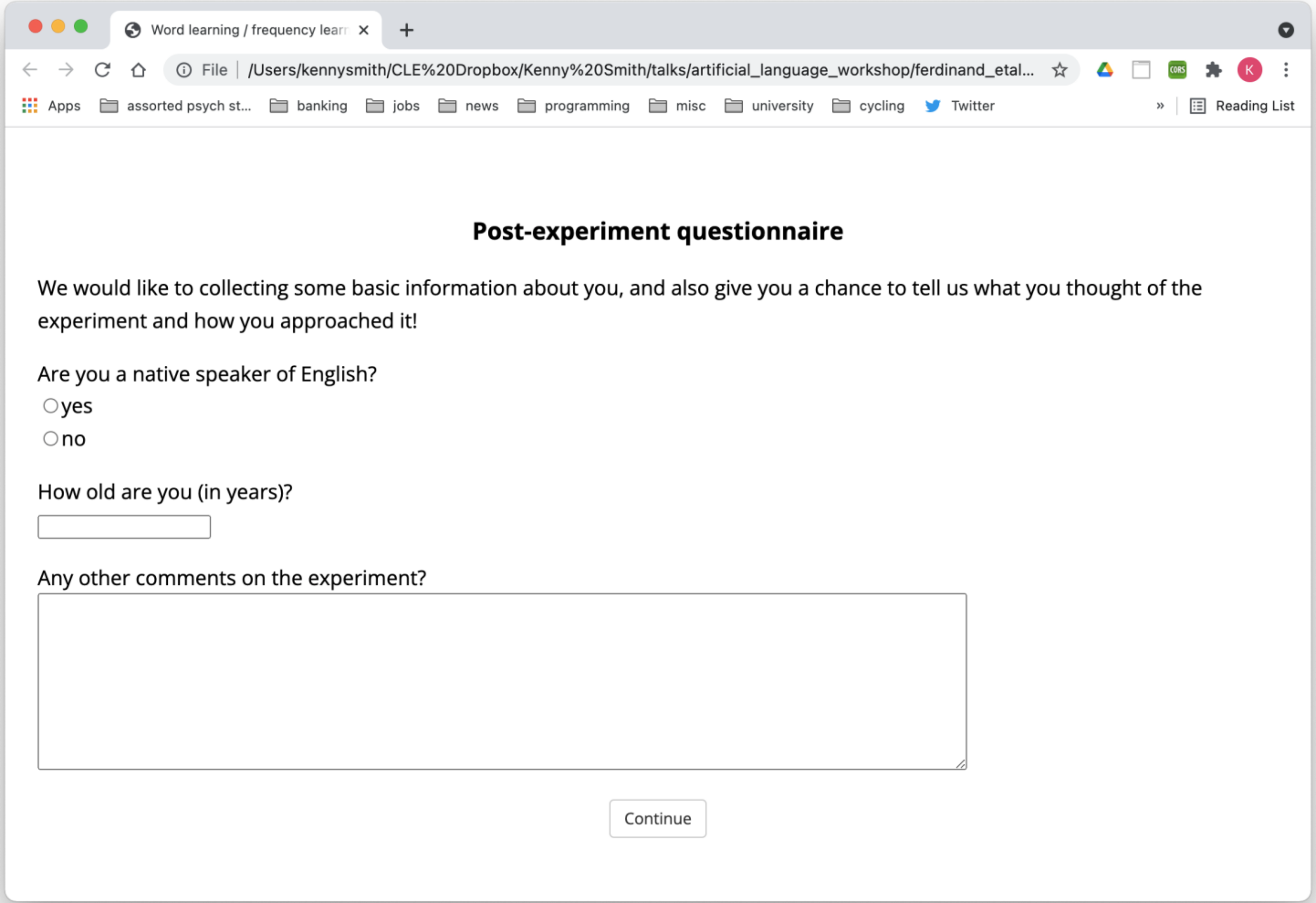
The “stimulus” is the text that will appear on the screen.
(I have added some formatting to make it look nice)

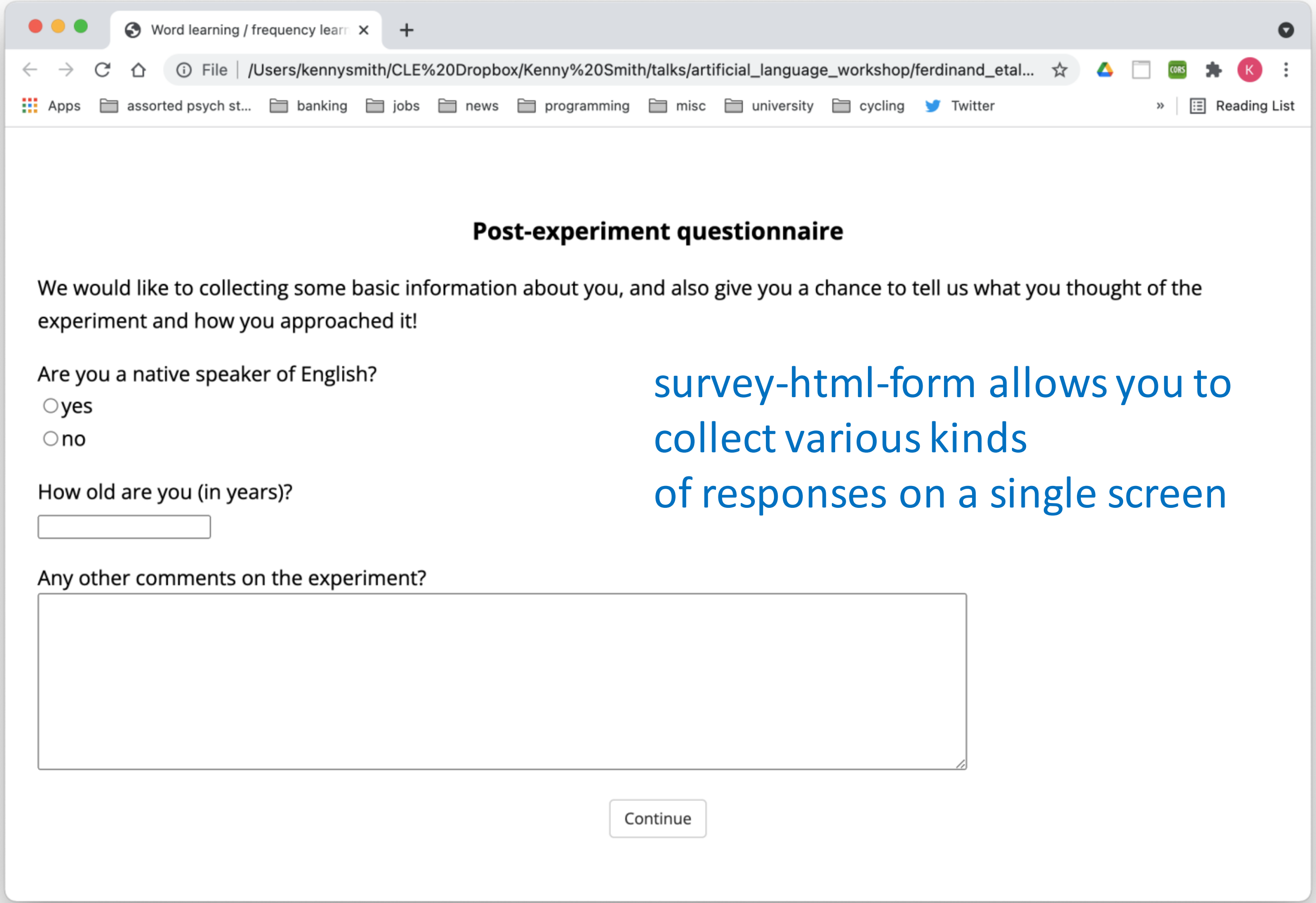
```
var consent_screen = {
  type: jsPsychHtmlButtonResponse,
  stimulus:
    "<h3>Welcome to the experiment</h3> \
  <p style='text-align:left'>Experiments begin with an information sheet that explains to the participant \
  what they will be doing, how their data will be used, and how they will be \
  remunerated.</p> \
  <p style='text-align:left'>This is a placeholder for that information, which is normally reviewed \
  as part of the ethical review process.</p>",
  choices: ["Yes, I consent to participate"],
};
```



Here there is only one choice, which is to consent

```
var consent_screen = {
  type: jsPsychHtmlButtonResponse,
  stimulus:
    "<h3>Welcome to the experiment</h3> \
  <p style='text-align:left'>Experiments begin with an information sheet that explains to the participant \
  what they will be doing, how their data will be used, and how they will be \
  remunerated.</p> \
  <p style='text-align:left'>This is a placeholder for that information, which is normally reviewed \
  as part of the ethical review process.</p>",
  choices: ["Yes, I consent to participate"],
};
```


Word learning / frequency learn
File
/Users/kennysmith/CLE%20Dropbox/Kenny%20Smith/talks/artificial_language_workshop/ferdinand_etal...
Apps
assorted psych st...
banking
jobs
news
programming
misc
university
cycling
Twitter
Reading List
Post-experiment questionnaire
We would like to collecting some basic information about you, and also give you a chance to tell us what you thought of the experiment and how you approached it!
Are you a native speaker of English?
yes
no
How old are you (in years)?
Any other comments on the experiment?
Continue



survey-html-form allows you to collect various kinds of responses on a single screen

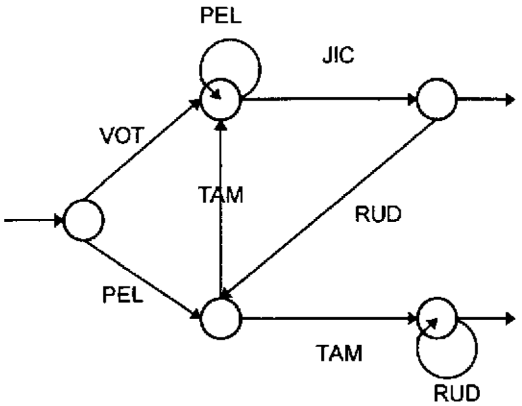
Demo – ferdinand_etal_2019/fks19.html

An example involving audio stimuli and keyboard responses

Based on Gomez, R. L. & Gerken, L. (1999) Artificial grammar learning by 1-year-olds leads to specific and abstract knowledge. *Cognition, 70,* 109-135.

Can participants implicitly learn an underlying grammar from presentation of positive examples, and subsequently differentiate between strings generated by / not generated by that grammar?

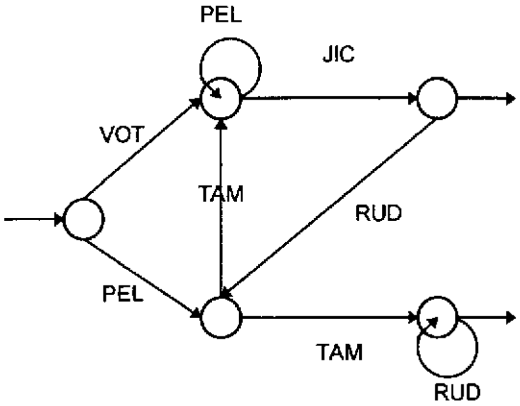
- Standard paradigm in the artificial grammar learning literature, e.g. Reber, 1967; Wilson et al., 2015; …



audio-keyboard-response plugin to present word sequences during training (NB no response required)

html-keyboard-response plugin to obtain yes/no responses on test trials

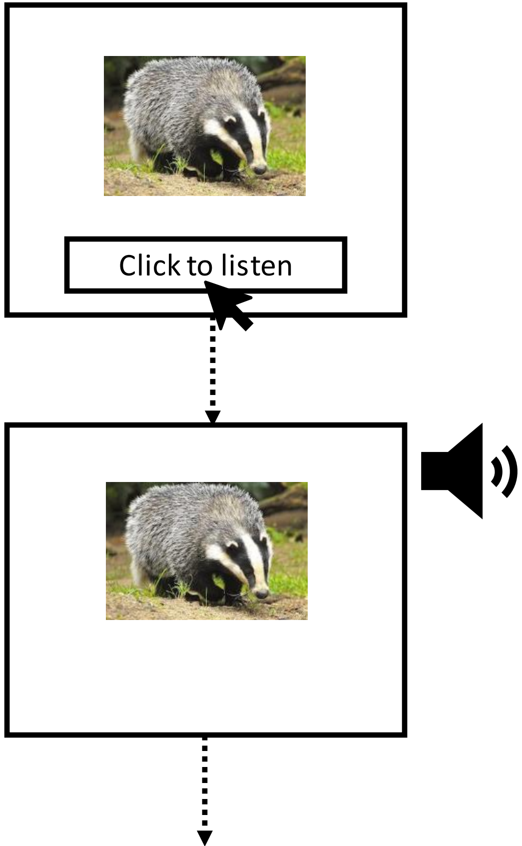
Demo – gomez_gerken_1999/gg99.html



An unexpected challenge: multimodal stimuli

Based on Martin, A., & White, J. (2021) Vowel harmony and disharmony are not equivalent in learning. *Linguistic Inquiry, 52,* 227-239.

Do participants assume that vowels should be harmonic or disharmonic?



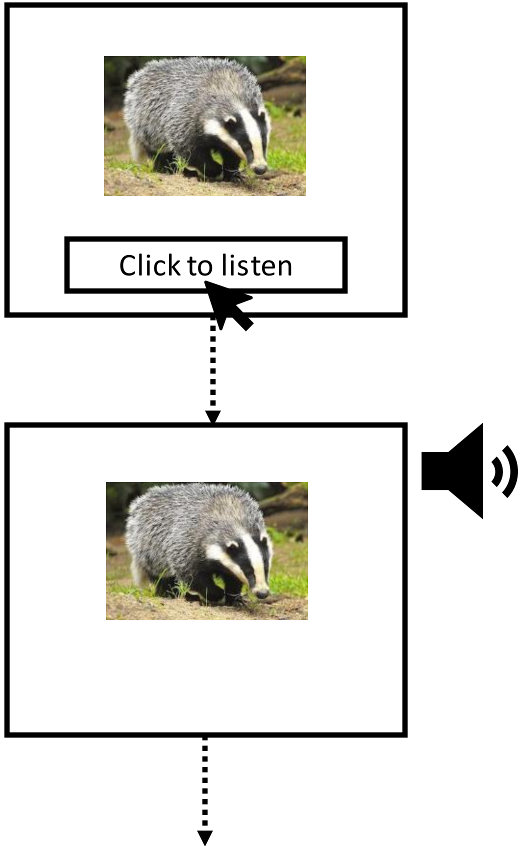
An unexpected challenge: multimodal stimuli

Previous examples were quite easy to achieve using existing plugins

But what if there isn't a plugin that does exactly what you want? E.g. image **and** audio on a single trial

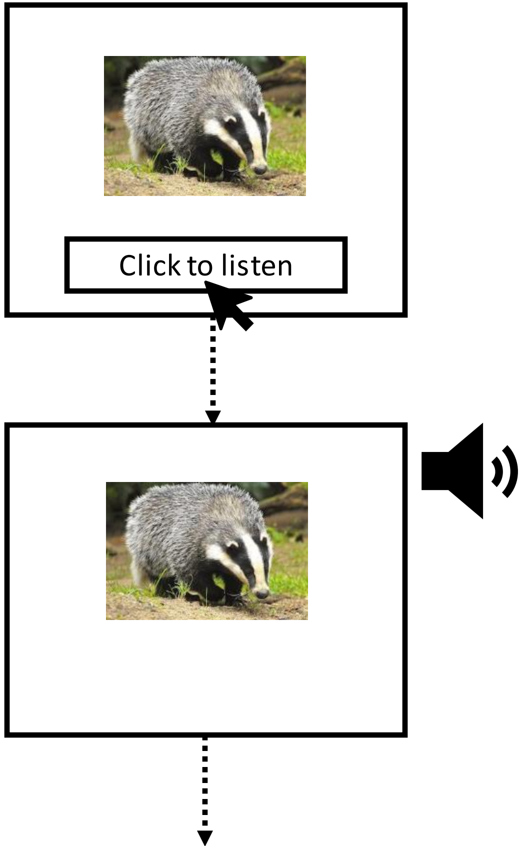
Two options:

- Smuggle in images to the prompt on an audio-button-response trial, or smuggle in audio to the prompt on an image-button response trial
- Edit the plugin to produce a custom plugin, e.g. an image-audio-button-response plugin!



An unexpected challenge: multimodal stimuli

```
var singular_trial = {
  type: jsPsychImageButtonResponse,
  stimulus: singular_image_filename,
  choices: ["Click to listen"],
};
var singular_trial_audio = {
  type: jsPsychImageAudioButtonResponse,
  image_stimulus: singular_image_filename,
  audio_stimulus: singular_label_filename,
  response_ends_trial: false,
  trial_ends_after_audio: true,
  choices: ["Click to listen"],
};
```



Demo – martin_white_2021/mw21.html

Additional things you will want to know how to do

- Random assignment to conditions
- Preloading stimuli
- Saving data
- Recording audio responses
- Reading in trial lists
- Contingent progression

Random assignment to conditions

```
//in a real experiment please give your conditions more transparent names than this!
var conditions = ["condition 1","condition 2","condition 3"]
var assigned_condition = jsPsych.randomization.sampleWithoutReplacement(conditions,1)[0]

var show_condition = {
  type: jsPsychHtmlButtonResponse,
  stimulus: "<p>Condition: " + assigned_condition + "</p>",
  choices:[]
};
```

Demo – other_examples/random_assignment.html

Preloading stimuli

- Downloading images, sound files etc takes an unpredictable amount of time.
- Trial durations ignore the time taken to download
- And beware: if you are testing your code, your browser might already have loaded and cached the stimuli, so you won't notice the lag

The solution: preloading

- Requires you to use the **preload** plugin
- Certain plugins mark parameters (e.g. stimuli) for automatic preloading by the preload plugin
- You can add everything else to a manual preload list

```
var preload = {
  type: jsPsychPreload,
  auto_preload: true,
};
```

See jsPsych pages on preloading: https://www.jspsych.org/7.3/overview/media-preloading/

Saving data

You need to save data on your server, not on the participant's computer!

Requires a bit of extra infrastructure: PHP script to write data to server (and a server that can run PHP – you will need your tech people to set this up)

See e.g.

- jsPsych pages on data: https://www.jspsych.org/7.3/overview/data/
- Week 7+ of my course (e.g. https://kennysmithed.github.io/oels2023/oels_practical_wk7.html)

You can save data at the end of the experiment or after every trial

- **Do trial-by-trial if you can!**

Recording audio responses

Accessing a participants' microphone (and camera!) is quite straightforward

- Although they have to approve, so you need to warn them you will request access

Recording audio data involves a little bit of extra work

- Another PHP script to save the audio data to your server
- See week 9 of my course: (https://kennysmithed.github.io/oels2023/oels_practical_wk9.html)

Audio quality varies, and sometimes the quality progressively degrades, but you get decent audio from 95% of participants

Reading in trial lists

I usually generate trial lists on the fly in the experiment code

- e.g. using a jsPsych.randomization.repeat, or a for-loop

If you prefer to have the trial list in a separate file on the server, that's possible too

- Reading in CSV files is slightly tricky because it involves a delay in accessing a file on the server (you have to force the browser to wait until the trial list is loaded)
- See weeks 9-10 of my course (https://kennysmithed.github.io/oels2023/oels_practical_wk9.html)

Contingent progression

Useful to be able to control who progresses beyond a certain point in your experiment

To ensure data quality, we included two important attention checks. First, near the end of the instructions, participants were told that the following page would ask for their Mechanical Turk ID, but that they should write "I understand" in the box instead. Second, we included four attention-check trials during the experiment itself (described in section 2.1.3). Participants who failed to enter "I understand" ($N = 59$) or who failed more than one attention trial ($N = 21$) were excluded and replaced. Participants were also excluded if they had participated in a pilot version of the experiment ($N = 4$).

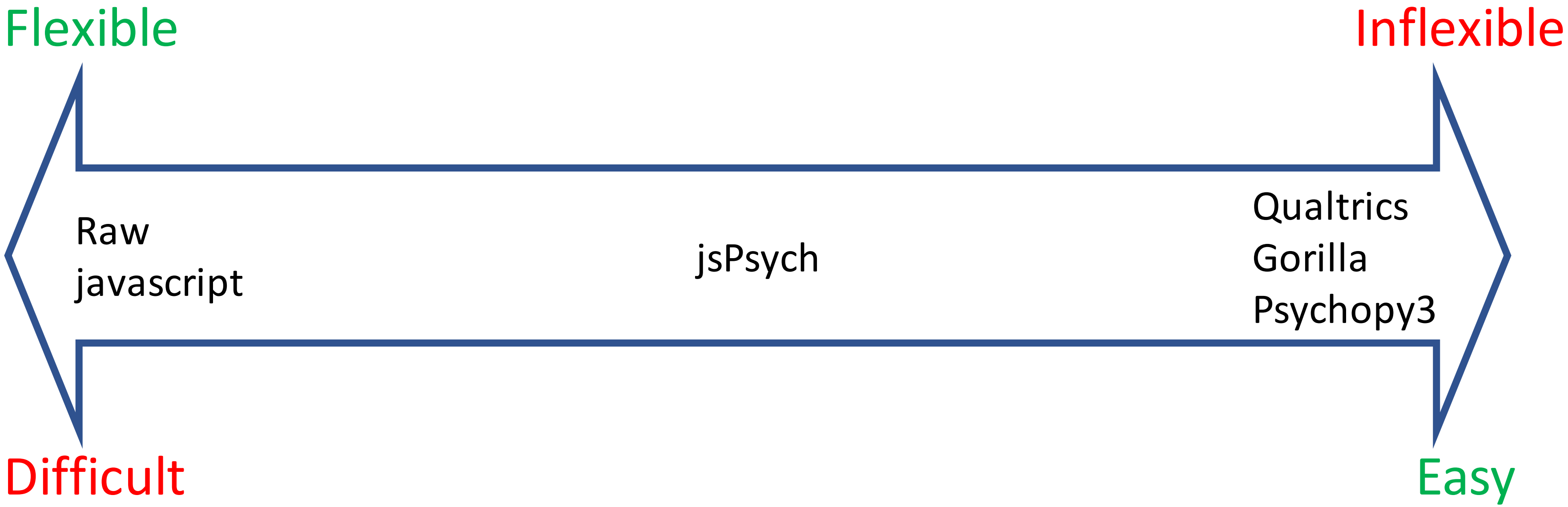
In total, we analyzed data from 120 participants (60 per condition), the number we preregistered. All participants (excluded or not) received compensation of US$6.

(Martin & White, 2021, p. 229-230)

See https://www.jspsych.org/7.3/overview/timeline/#conditional-timelines

Demo – other_examples/contingent_progression.html

Other platforms are available!



Questions?

References

Culbertson, J., Smolensky, P., & Legendre, G. (2012). Learning biases predict a word order universal. *Cognition, 122,* 306–329.

de Leeuw, J. R. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a web browser. *Behavior Research Methods*, *47*, 1-12. doi:10.3758/s13428-014-0458-y.

Difallah, D., Filatova, E., & Ipeirotis, P. (2018). Demographics and Dynamics of Mechanical Turk Workers. In *Proceedings of WSDM 2018: The Eleventh ACM International Conference on Web Search and Data Mining*.

Enochson, K., & Culbertson, J. (2015). Collecting Psycholinguistic Response Time Data Using Amazon Mechanical Turk. *PLoS ONE, 10,* e0116946.

Ferdinand, V., Kirby, S., & Smith, K. (2019). The cognitive roots of regularization in language. *Cognition, 184,* 53-68.

Gomez, R. L. & Gerken, L. (1999) Artificial grammar learning by 1-year-olds leads to specific and abstract knowledge. *Cognition, 70,* 109-135.

Hudson Kam, C., & Newport, E. L. (2005). Regularizing unpredictable variation: The roles of adult and child learners in language formation and change*. Language Learning and Development, 1,* 151–195.

Loy, J. E., & Smith, K. (2021). Speakers Align With Their Partner's Overspecification During Interaction. *Cognitive Science, 45,* e13065.

Martin, A., & White, J. (2021) Vowel harmony and disharmony are not equivalent in learning. *Linguistic Inquiry, 52,* 227-239.

Monroe, R. et al. (2010). Crowdsourcing and language studies: the new generation of linguistic data. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science, 349,* aac4716.

Pavlick, E. et al. (2014). The Language Demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics, 2,* 79-92.

Reali, F., & Griffiths, T. L. (2009). The evolution of frequency distributions: Relating regularization to inductive biases through iterated learning. *Cognition, 111,* 317–328.

Reber, A.S. (1967) Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behavior, 6,* 855–863.

Smith, K., & Wonnacott, E. (2010). Eliminating unpredictable variation through iterated learning. *Cognition, 116,* 444-449.

Sprouse, J. (2011). A validation of Amazon Mechanical Turk for the collection of acceptability judgments in linguistic theory. *Behavior Research Methods, 43,* 155-167.

Vouloumanos, A. (2008). Fine-grained sensitivity to statistical information in adult word learning. *Cognition, 107,* 729–742

Wilson, B., Smith, K., & Petkov, C. (2015). Mixed-complexity artificial grammar learning in humans and macaque monkeys: Evaluating learning strategies. *European Journal of Neuroscience, 41,* 568-578.