

Building linguistic experiments in PsychoPy: Identification task

Department of Linguistics

University of Toronto

Oct 2017

Na-Young Ryu

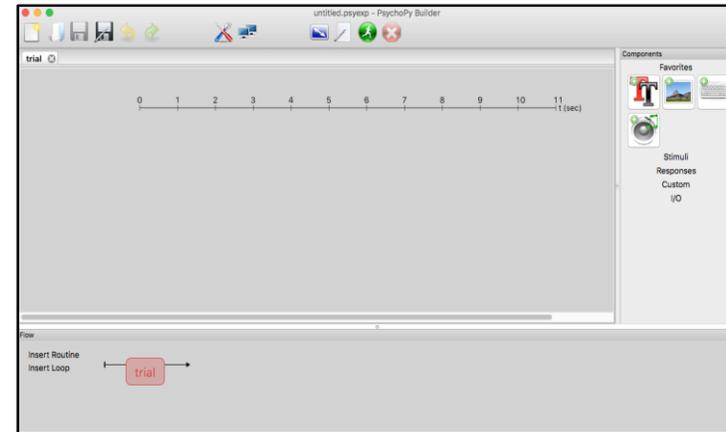
Building an experiment is like a building a house!

- Building an experiment using PsychoPy is like a building a house.

- ① Plan.
- ② Decide how many rooms.
- ③ Create rooms by adding furniture.
- ④ Hold a house warming party.



- ① Plan.
- ② Create *Routines* and *Loops in Flow*.
- ③ Add *Components in Routine*.
- ④ Run an experiment.



PsychoPy *builder view*

- What you need is to open the **PsychoPy builder view** and follow our step-by step tutorial on how to build an experiment. You will learn to create an experiment on the PsychoPy builder view with little-to-no experience in programming.
- If you have any questions about our tutorials, e-mail Na-Young Ryu at nayoung.ryu@mail.utoronto.ca

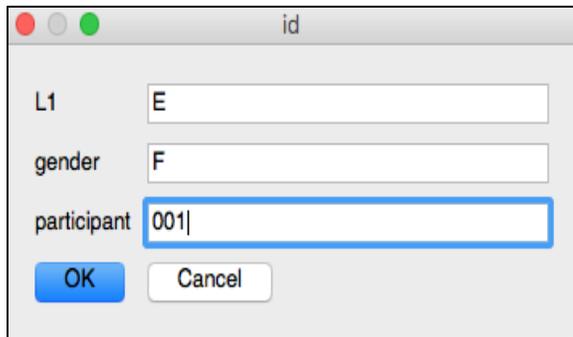
Download our materials for your experiments

- **Why build experiments using the PsychoPy builder?**
 - It is free and easy to generate a wide range of linguistic experiments in the *Builder* view.
 - The builder view allows you to create experiments visually, so the Python programming language is not needed.
 - Output data is automatically generated after running an experiment (e.g. xlsx, csv, text file)
 - The types of experiment, along with manuals we provide, are as follows:
 - **Word production task (auto)** - each word appears every 3 seconds.
 - **Word production task (keyboard response)** - each word appears after a participant presses the spacebar.
 - **Identification task** - participants listen to a sound and identify it among several sounds.
 - **AX discrimination task** - participants listen to two sounds and determine whether they are the same or not.
 - **Rating task** - Participants listen to sounds and rate the accuracy of sounds on a scale of 1 - 7.
 - The materials we offer are on based on **PsychoPy 1.85.2**.
 - You can install PsychoPy on your computer by downloading it from www.psychopy.org.
- **Where to download our materials for linguistic experiments**
 - We are very happy to share our linguistic experiments using the PsychoPy builder, so feel free to modify them for your own experimental purposes.
 - To download our materials for linguistic experiments, please go to:
<http://individual.utoronto.ca/rrrnny/experiments.html>

Identification Task: Goals & Procedures

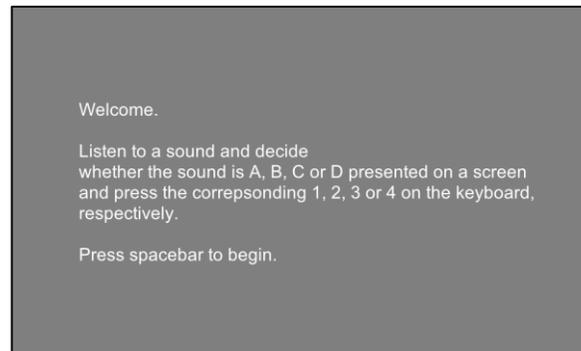
- **Goals:** We provide instructions on how to create an identification experiment using the PsychoPy Builder view (Peirce 2007). This experiment is designed to ask participants to listen to a sound and identify it among several sounds.
- **Procedures:** Show **Instruction 1** followed by **trial1** to show auditory stimuli and identify for the practice session; then you can use the same basic structure for **Instruction2** followed by **trial2** for the main session and the “**thank you**” message to participants at the end.

1. Filling out participant information

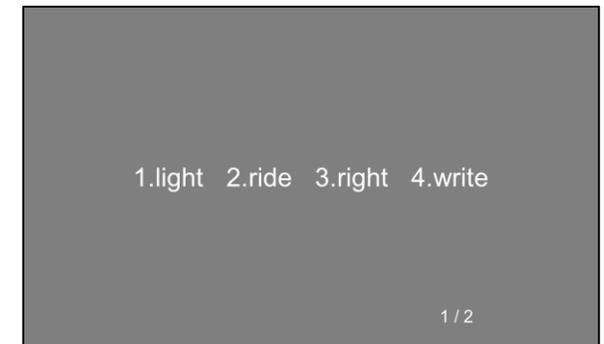


A screenshot of a PsychoPy Builder dialog box titled "id". It contains three input fields: "L1" with the value "E", "gender" with the value "F", and "participant" with the value "001". There are "OK" and "Cancel" buttons at the bottom.

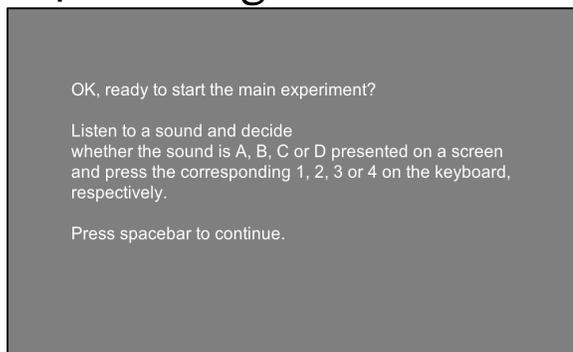
2. Showing instructions



3. Displaying stimuli and getting responses



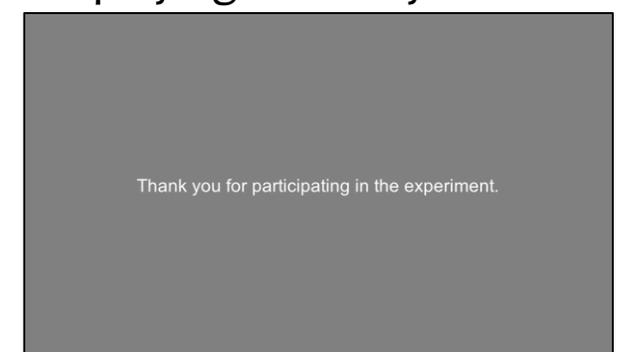
4. Showing instructions



5. Displaying stimuli and getting responses



6. Displaying “Thank you” message



How to build the identification task?

- **Step1:** Create a folder to store your experiment files
 - We have already provided two Excel files containing word lists for the rating task.
 - Please find them in the folder named “Id Task”: “main_stimuli” and “practice_stimuli”.
- **Step2:** Alter your Experiment Settings (e.g. participant ID, format of output file).
- **Step3:** Create your *Routines* (e.g. instructions, trials).
- **Step4:** Add the *Routines* and *Loops* to the *Flow*.
- **Step5:** Add *Components* to the *Routine*.
- **Step6:** Run an experiment.

Step 1: Excel file with sound files

w1	w2	w3	w4	sound	corrAns
side	site	sort	size	sound/side.wav	1
tide	tight	though	think	sound/tight.wav	2
light	ride	right	write	sound/ride.wav	2
stop	top	cop	pop	sound/top.wav	2

Step 2: Participant info

id

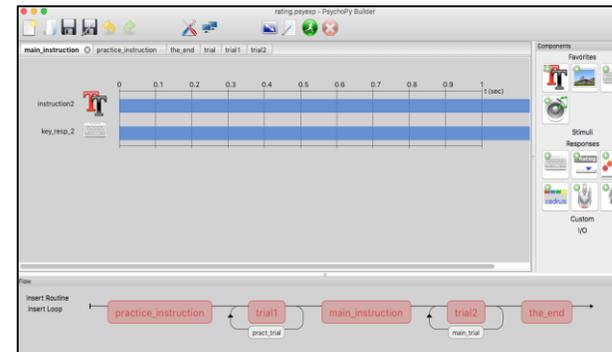
L1: E

gender: F

participant: 001

OK Cancel

Step 3-5: Create Routines and loops



Step 6 : Run an experiment

Welcome.

Listen to a sound and decide whether the sound is A, B, C or D presented on a screen and press the corresponding 1, 2, 3 or 4 on the keyboard, respectively.

Press spacebar to begin.

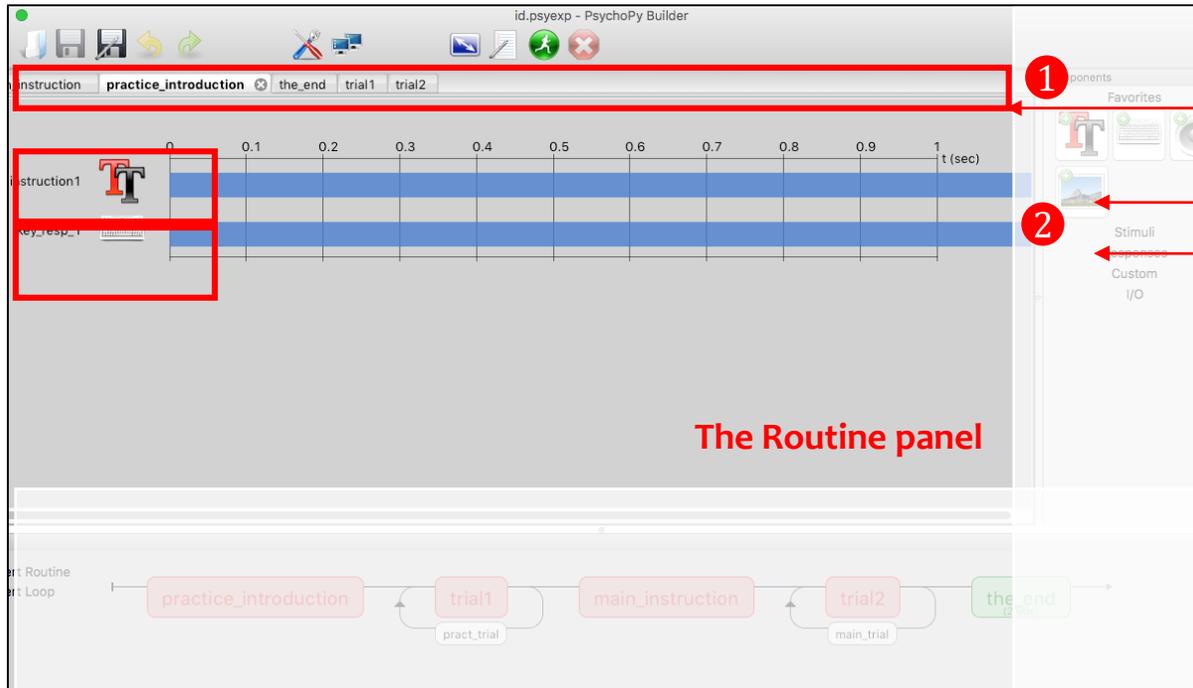
PsychoPy *Builder* interface

The screenshot displays the PsychoPy Builder interface for a file named 'id.psyexp'. The top toolbar includes icons for file operations (save, print, undo, redo), editing (wrench, eraser), and execution (run, stop). Below the toolbar, a tabbed interface shows 'practice_introduction' as the active routine. The main workspace is divided into three panels:

- 1 The Routine panel:** A timeline from 0 to 1 second. Two components are visible: 'instruction1' (represented by a large 'T' icon) and 'key_resp_1' (represented by a keyboard icon). Both components are active throughout the 1-second duration.
- 2 The Flow panel:** A sequence of routine blocks: 'practice_introduction' (red), 'trial1' (red) with a sub-block 'pract_trial' (white), 'main_instruction' (red), 'trial2' (red) with a sub-block 'main_trial' (white), and 'the_end (2.00s)' (green). Arrows indicate the flow from left to right, with loops between 'trial1' and 'main_instruction', and between 'trial2' and 'main_instruction'.
- 3 The Component panel:** A sidebar on the right containing 'Favorites' (with icons for text, keyboard, and image), and categories for 'Stimuli', 'Responses', 'Custom', and 'I/O'.

- The PsychoPy builder view comprises three panels: (1) **Routines**, (2) **Flow**, and (3) **Component panel**.

The *Routine* Panel



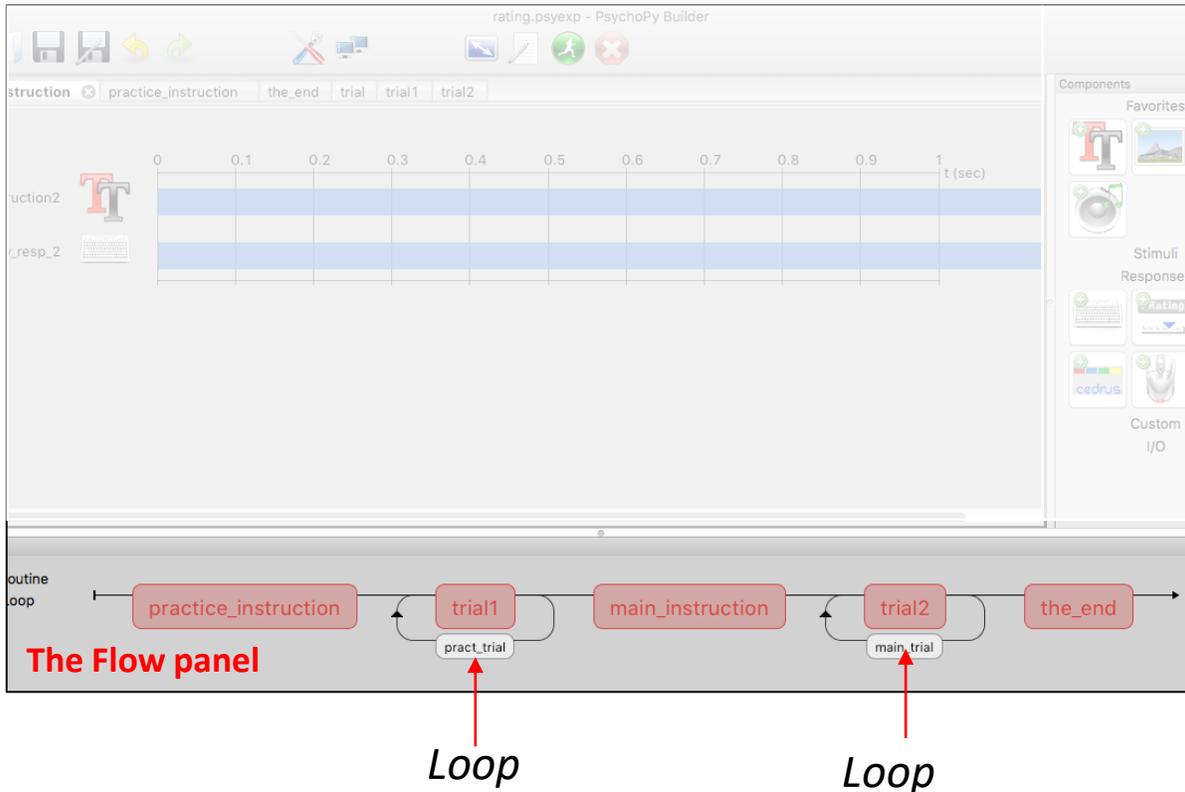
The screenshot displays the PsychoPy Builder interface. At the top, a tab bar contains five tabs: 'instruction', 'practice_introduction', 'the_end', 'trial1', and 'trial2'. A red box labeled '1' highlights these tabs, with an arrow pointing to the text 'These tabs show Routines.'. Below the tabs is a timeline from 0 to 1 second. Two components are visible: 'struction1' (a text component) and 'key_resp_1' (a keyboard component), both highlighted with red boxes and a red box labeled '2'. An arrow points from the text 'Text component' to the 'struction1' box, and another arrow points from 'Keyboard component' to the 'key_resp_1' box. The text 'The Routine panel' is written in red in the center of the timeline area. At the bottom, a flowchart shows the sequence of routines: 'practice_introduction' (orange), 'trial1' (orange) with a sub-loop 'pract_trial', 'main_instruction' (orange), 'trial2' (orange) with a sub-loop 'main_trial', and 'the_end' (green).

① An experiment can have **many Routines**. This rating task has **five Routines**.

You can switch between your *Routines* by selecting the different tabs.

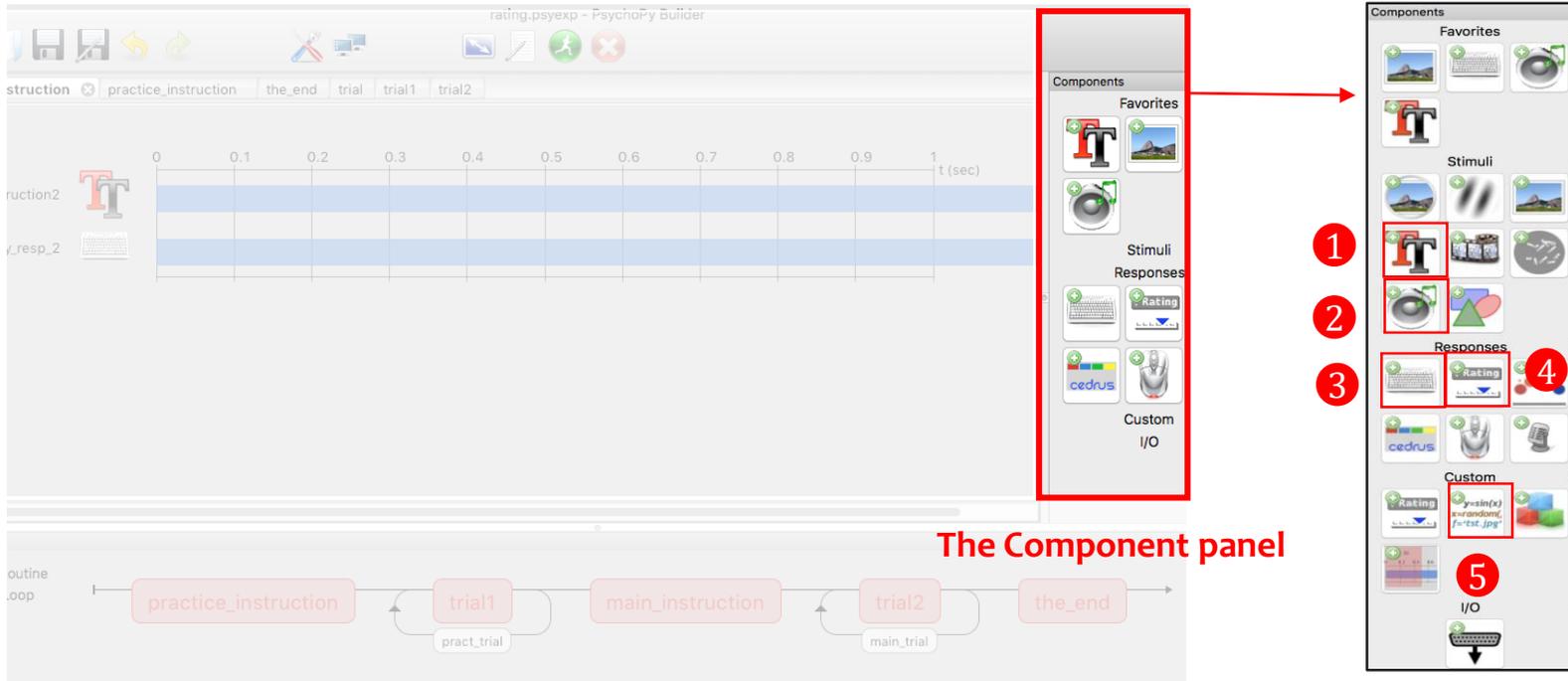
② The Routine shown here has a **Text** and **Keyboard component**.

The *Flow* Panel



- All experiments have exactly one **Flow**.
- The experiment proceeds from left to right, and each part of the *Flow* panel is executed in turn. That is, everything is run in the order in which it appears from left to right.
- The *Flow* can contain **Loops** controlling how a **Routine** (e.g. a trial) is repeated, both in terms of how many repeats are made and how things change from one trial to the next.

The *Components* panel



The Component panel

The PsychoPy builder view offers various components. Commonly-used components for linguistic experiments are as follows:

- 1 **Text Component** - Display text on the screen.
- 2 **Sound Component** - Play sounds.
- 3 **Keyboard Component** - Receive input from the keyboard.
- 4 **RatingScale Component** - Collect a numeric rating or a choice from a few alternatives, via the mouse, the keyboard or both.
- 5 **Code Component** - Insert short pieces of python code into your experiments (e.g. time stamp for the production task)

Names for the identification task

- **Everything in a PsychoPy experiment needs a unique name.**
- E.g. if you have a *Routine* called 'pract_introduction' you can't have a *Text* component called 'pract_introduction'
- The name must:
 - contain only letters, numbers and underscores.
 - not contain spaces, punctuation or mathematical symbols.

Routines	pract_introduction	trial1	main_introduction	trial2	the_end
Loops		pract_trial		main_trial	
Text component	instruction1	word1	instruction2	word2	text
		trial_number1		trial_number2	
Sound component		sound_1		Sound_2	
Keyboard component	key_resp_1	Key_resp_2	key_resp_3	Key_resp_4	

Step1: Create Excel files with your sound list.



main_stimuli.xlsx

w1	w2	w3	w4	sound	corrAns
side	site	sort	size	sound/side.wav	1
tide	tight	though	think	sound/tight.wav	2
light	ride	right	write	sound/ride.wav	2
stop	top	cop	pop	sound/top.wav	2

Sound files are in the fold "sound"

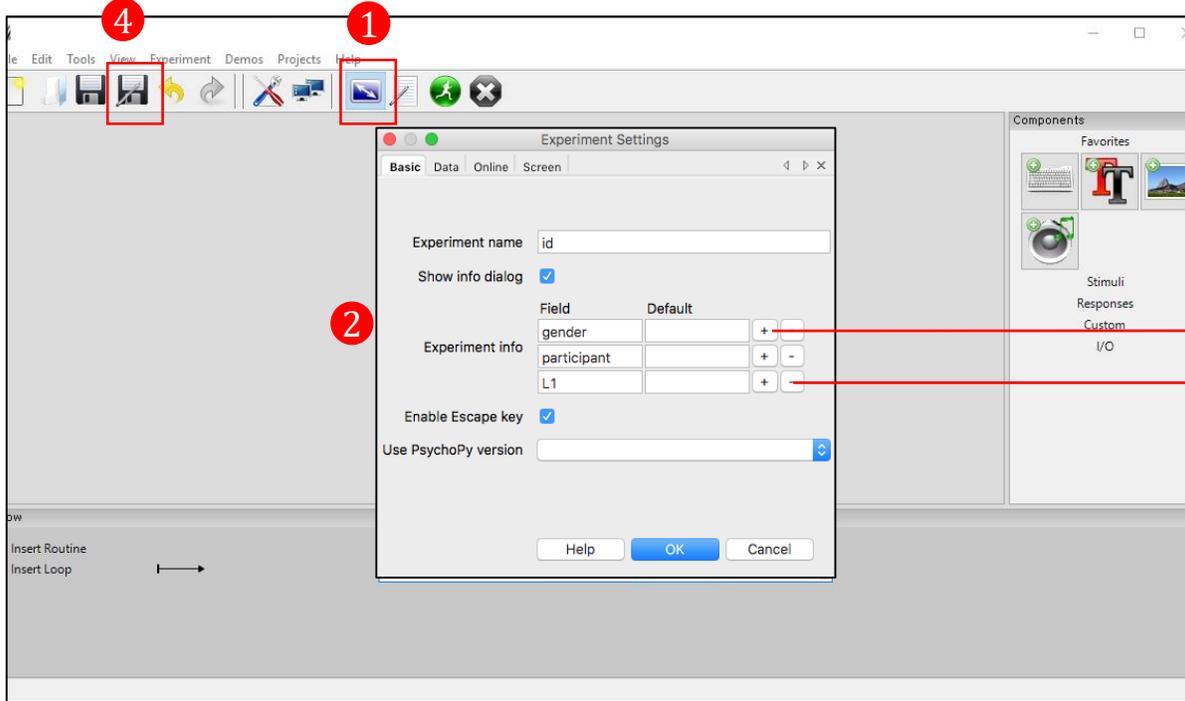


practice_stimuli.xlsx

w1	w2	w3	w4	sound	corrAns
light	ride	right	write	sound/ride.wav	2
stop	top	cop	pop	sound/top.wav	2

- Create a folder named "Identification Task" on your computer to store experiment files.
- In the identification task, we need two Excel files with names of words and auditory stimuli/sound files. ('main_stimuli.xlsx' and 'practice_stimuli.xlsx')
- We also need to create a folder named "sound" to store auditory stimuli.
- Finally, create a folder "data" in which an output file of this experiment will be automatically generated.

Step2: Alter experiment settings

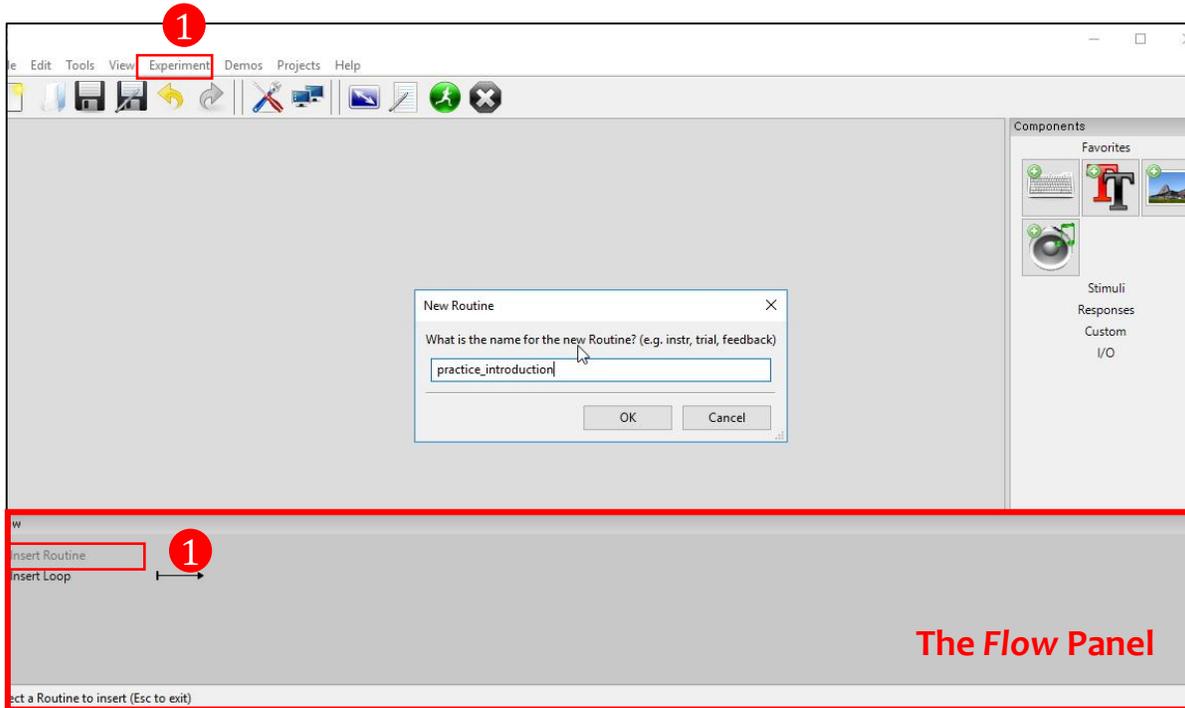


 To add experiment info
 To delete experiment info



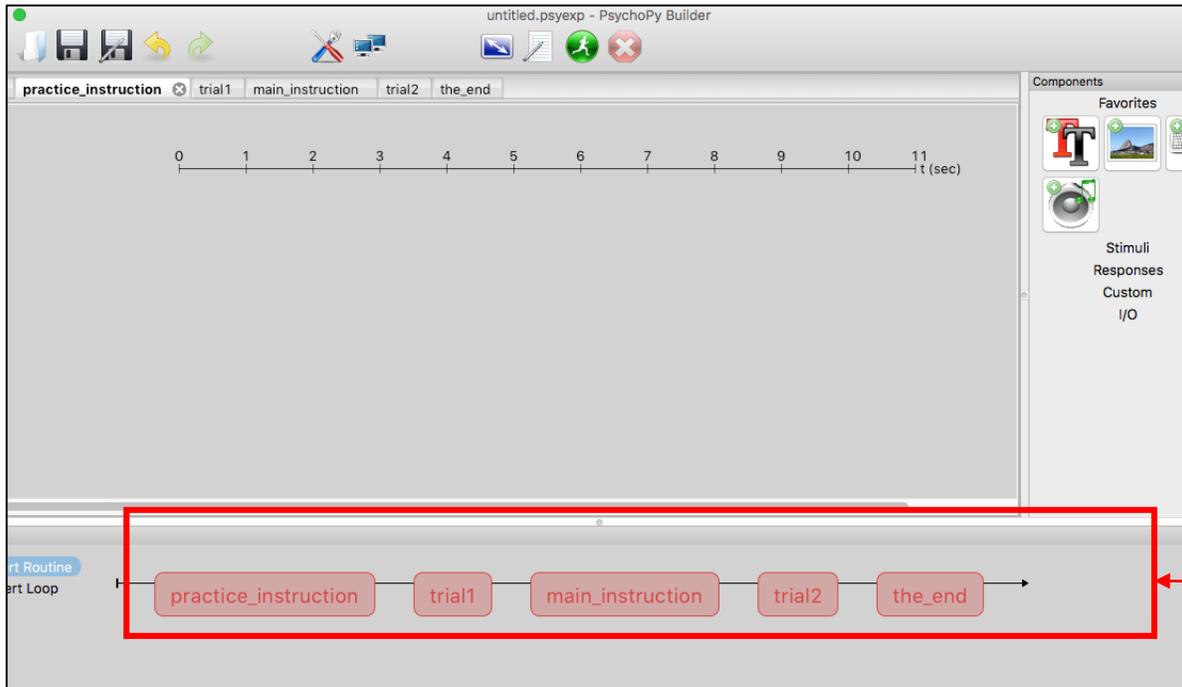
- 1 Click on  on the top of the screen for **experiment settings**.
- 2 Write an experiment name, e.g. “id”, and modify the experiment information (e.g. participant ID, gender, L1) by clicking +/- icons.
- 3 If you would like to change the forms of output files or the size of the window, go to the *Data* or *Screen* tab.
- 4 Save the PsychoPy file named “**Identification Task.psyexp**” in the same folder after setting up the rating task.

Step3: Create *Routines* in the *Flow*



- 1 Click on “Insert Routine” in the *Flow* panel to insert a new *Routine* and write down the name of the *new Routine*, or use the menu bar: Experiment > Insert *Routine* in *Flow* > New *Routine*.

Step4: Add the *Routines* to the *Flow*



Insert five *Routines* in the *Flow* panel

- Multiple *Routines* can then be combined in the *Flow* panel, which controls the order in which these occur and the way in which they repeat. For this rating task, **five** *Routines* in the *Flow* panel were inserted: **(1) Practice instruction, (2) trial1, (3) Main instruction, (4) trial2, and (5) “thank you” message** at the end.
- These are combined in the *Flow* panel so that the **practice instruction** comes first, followed by **trial1**, followed by the **main instruction** and **trial2**, and finally the **“thank you”** *Routine* comes last.

Step4: Insert *Loops* in the *Flow*

Flow diagram showing a sequence of routines: practice_introduction, trial1, main_instruction, trial2, and the_end. Two loops are shown: one around trial1 and pract_trial, and another around trial2 and main_trial. A red box highlights the 'Insert Loop' button in the 'Insert Routine' panel, with a red circle containing the number 1 next to it. Red arrows point from the text 'Loop' below to the two loop structures in the flow diagram.

pract_trial Properties dialog box. The 'loopType' is set to 'random'. The 'nReps' field is set to 1. The 'Conditions' field is set to 'practce_stimuli.xlsx'. A red box highlights the 'nReps' and 'Conditions' fields. A red arrow points from the text 'No of the repetition' to the 'nReps' field. Another red arrow points from the text 'When inserting a loop into the Flow you can browse to find the file you wish to use for this.' to the 'Conditions' field.

w1	w2	w3	w4	sound	corrAns
light	ride	right	write	sound/ride.wav	2
stop	top	cop	pop	sound/top.wav	2

6 parameters
2 conditions

Loops control the repetition of Routines and the choice of stimulus parameters for each.

1. Click on the “insert loop” button in the *Flow* panel to insert a loop and select the points where you want to start/finish.
 - Loops can span across multiple *Routines*.
 - Loops can nest (you can have loops around loops)

Loops and Routines can also be edited or removed from the *Flow* panel by clicking or right-clicking.

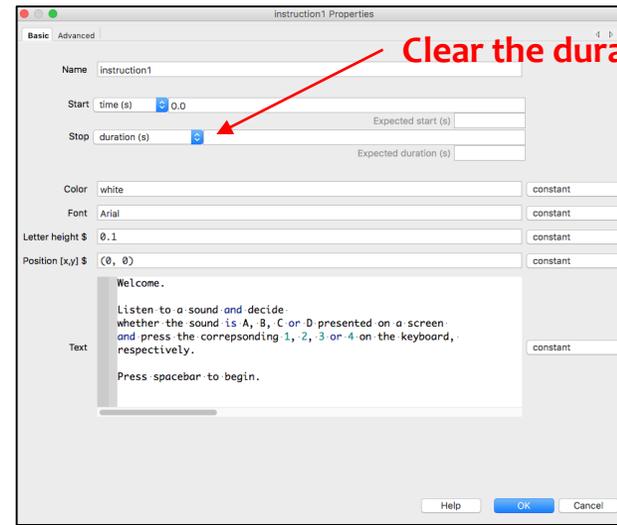
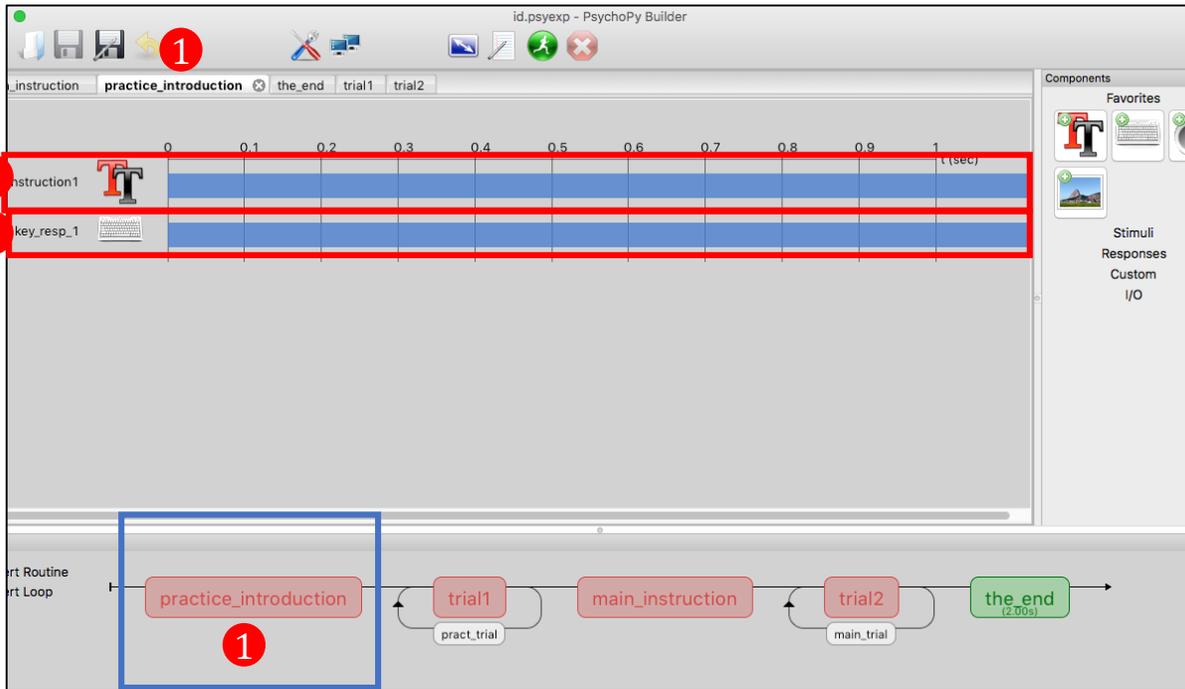
Common settings used by *Components*

The screenshot shows the 'instruction1 Properties' dialog box with the following settings:

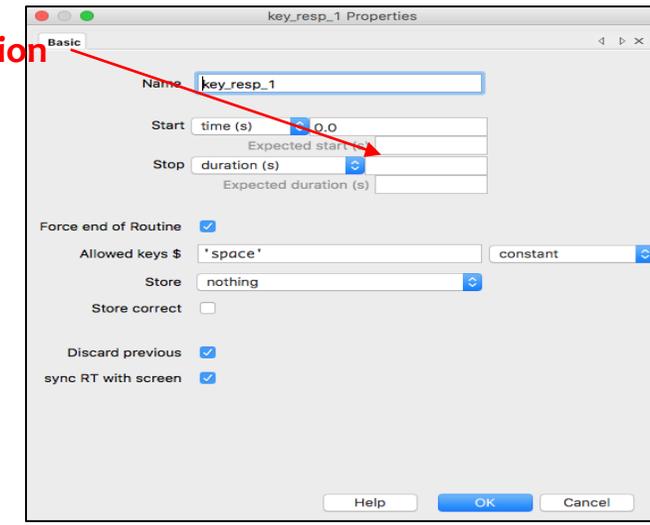
- Name:** instruction1
- Start:** time (s) 0.0
- Stop:** duration (s)
- Color:** white
- Font:** Arial
- Letter height \$:** 0,1
- Position [x,y] \$:** (0, 0)
- Text:** welcome.
Please read each word as it appears on the screen.
Press spacebar to begin.

- 1 Start time (seconds)** – the time after the start of the *Routine* that the stimulus will first appear.
- 2 Duration (seconds)** – the length of time it will last. **If blank then the Component will go on indefinitely.**
- 3 Font:** default font is “Arial”. If you have font issues, please try different fonts such as “Times New Roman” and “Fira Sans”.
- 4 Position** – the position on the screen in the given unit. **[0,0] is the centre of the screen.**

Step 5: Add *Components* to the *Routine* [practice_instruction]



2 Text properties

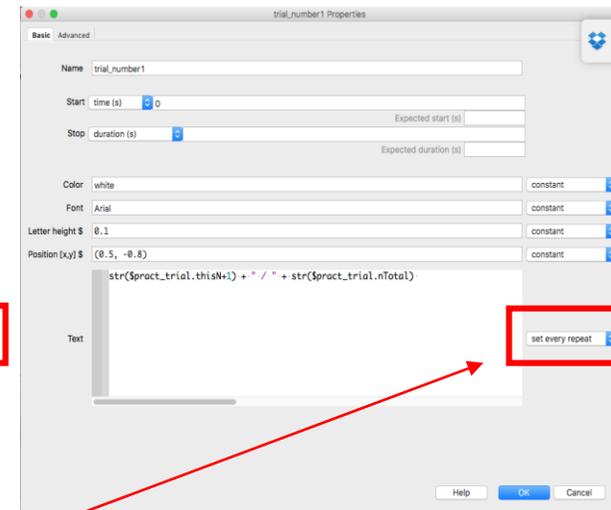
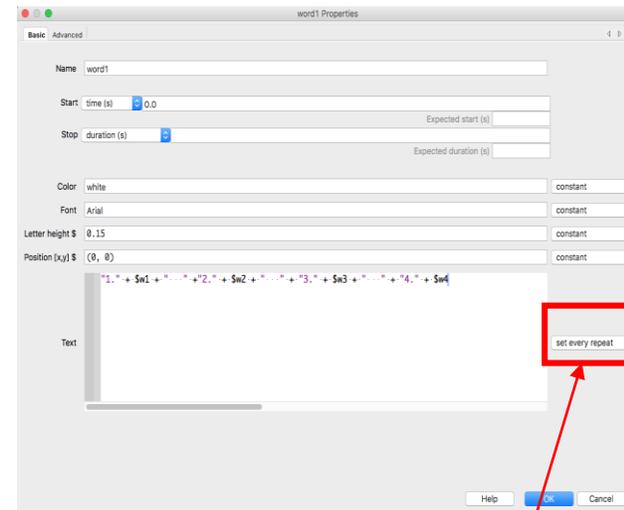
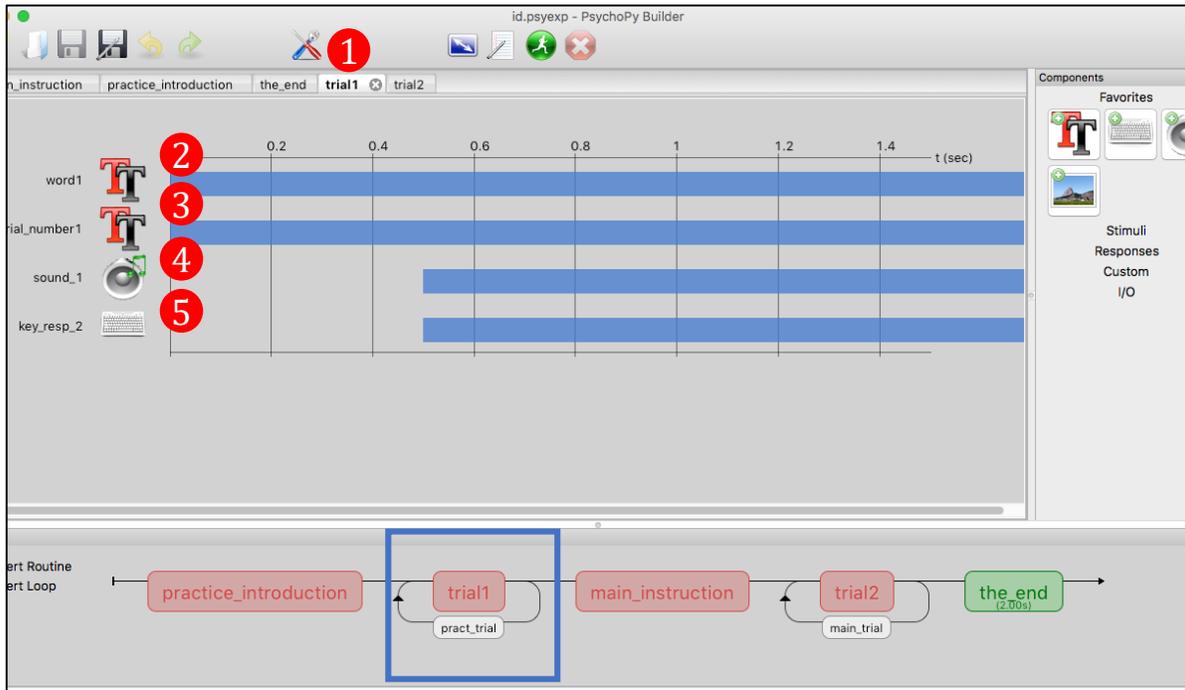


3 Keyboard properties

For the *Routine* [practice_instruction], a *Text* component and a *Keyboard* component are needed.

- 1 Select the *Routine* [practice_instruction] in the tab bar or in the *Flow* panel.
- 2 Click on  in the *Component* panel and fill out the *Text* properties: [In the *Basic* tab] Font: Arial, Color: white, Text: an instruction message is inserted, [In the *Advanced* tab] **Wrap width:\$ 2**
- 3 Click on  in the *Component* panel and fill out the *Keyboard* properties: Allowed key :\$ 'space', Store: nothing.

Step5: Add *Components* to the *Routine* [Trial1]



- ② Text properties (word1)
- ③ Text properties(trial_number1)

'set every repeat'
It will be updated every repeat of the Routine.

For the Routine [trial1], two Text components, a Sound component and a Keyboard component are needed.

- ① Select [trial1] in the tab, or in the Flow panel.
- ② Click on  in the Component panel and fill out the text properties: Stop duration: blank, Font: Arial, Letter height: 0.15, Position: (0.0), Text: "1." + \$w1 + " " + "2." + \$w2 + " " + "3." + \$w3 + " " + "4." + \$w4
- ③ Click on  in the Component panel and fill out the text properties: Font: Arial, Letter height: 0.1, Position:(0.5, -0.8) Text: `str($pract_trial.thisN+1) + " / " + str($pract_trial.nTotal)`

Step5: Add *Components* to the *Routine* [Trial1]

The screenshot shows the PsychoPy Builder interface. At the top, the routine name is 'id.psyexp - PsychoPy Builder'. Below it, there are tabs for 'practice_introduction', 'the_end', 'trial1', and 'trial2'. The main area is a timeline from 0 to 1.4 seconds. Components are placed on the timeline: 'word1' (red 'T' icon) at 0.2s, 'trial_number1' (red 'T' icon) at 0.3s, 'sound_1' (speaker icon) at 0.5s, and 'key_resp_2' (keyboard icon) at 0.6s. Red circles with numbers 2, 3, 4, and 5 are placed next to these components. Below the timeline is a flowchart showing the routine structure: 'practice_introduction' leads to a loop 'trial1' (containing 'pract_trial'), which leads to 'main_instruction', then to another loop 'trial2' (containing 'main_trial'), and finally to 'the_end (2.00s)'. A blue box highlights the 'trial1' loop in the flowchart.

The screenshot shows the 'sound_1 Properties' dialog box. The 'Name' field is 'sound_1'. The 'Start' field is 'time (s)' with a value of 0.5. The 'Stop' field is 'duration (s)'. The 'Sound' field is '\$sound'. The 'Repeat' dropdown menu is set to 'set every repeat'. The 'Volume' field is '1' and the 'Volume' dropdown is set to 'constant'. There are 'Help', 'OK', and 'Cancel' buttons at the bottom.

4 Sound properties (sound_1)

The screenshot shows the 'key_resp_2 Properties' dialog box. The 'Name' field is 'key_resp_2'. The 'Start' field is 'time (s)' with a value of 0.5. The 'Stop' field is 'duration (s)'. The 'Allowed keys' field is '"1","2","3","4"'. The 'Store' dropdown menu is set to 'last key'. The 'Store correct' checkbox is checked. The 'Correct answer' field is 'corrAns'. The 'Discard previous' checkbox is unchecked. The 'sync RT with screen' checkbox is checked. There are 'Help', 'OK', and 'Cancel' buttons at the bottom.

5 Keyboard properties (key_resp_2)

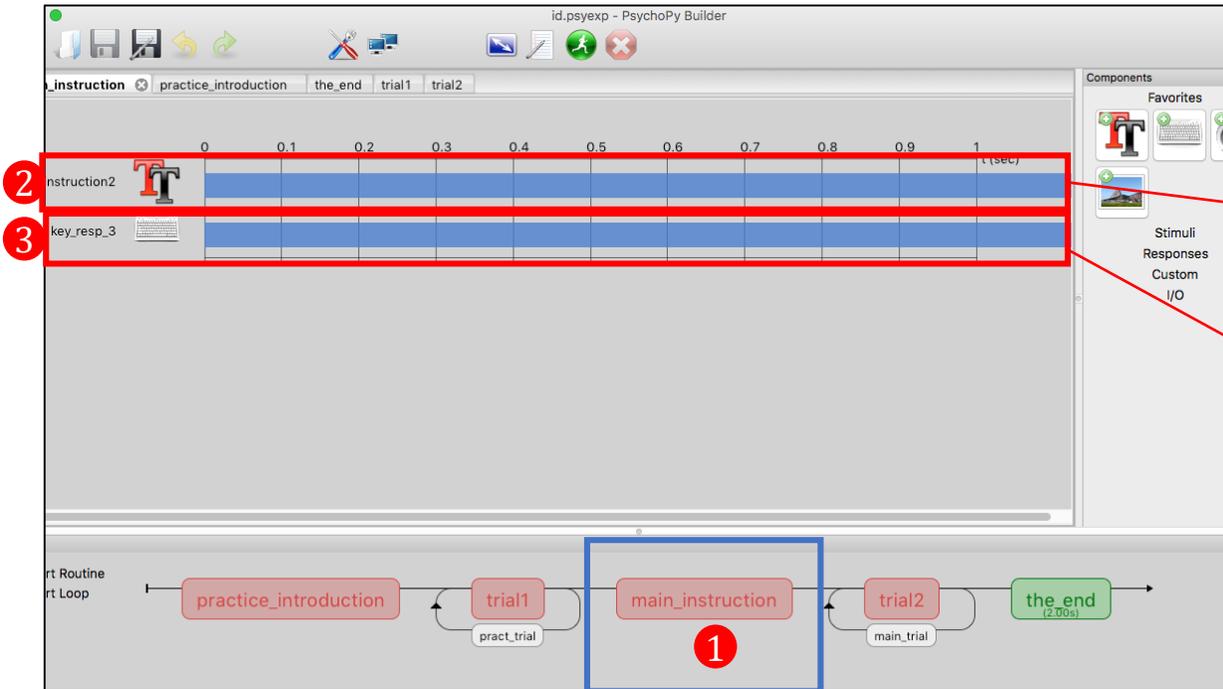
'set every repeat'

It will be updated every repeat of the Routine.

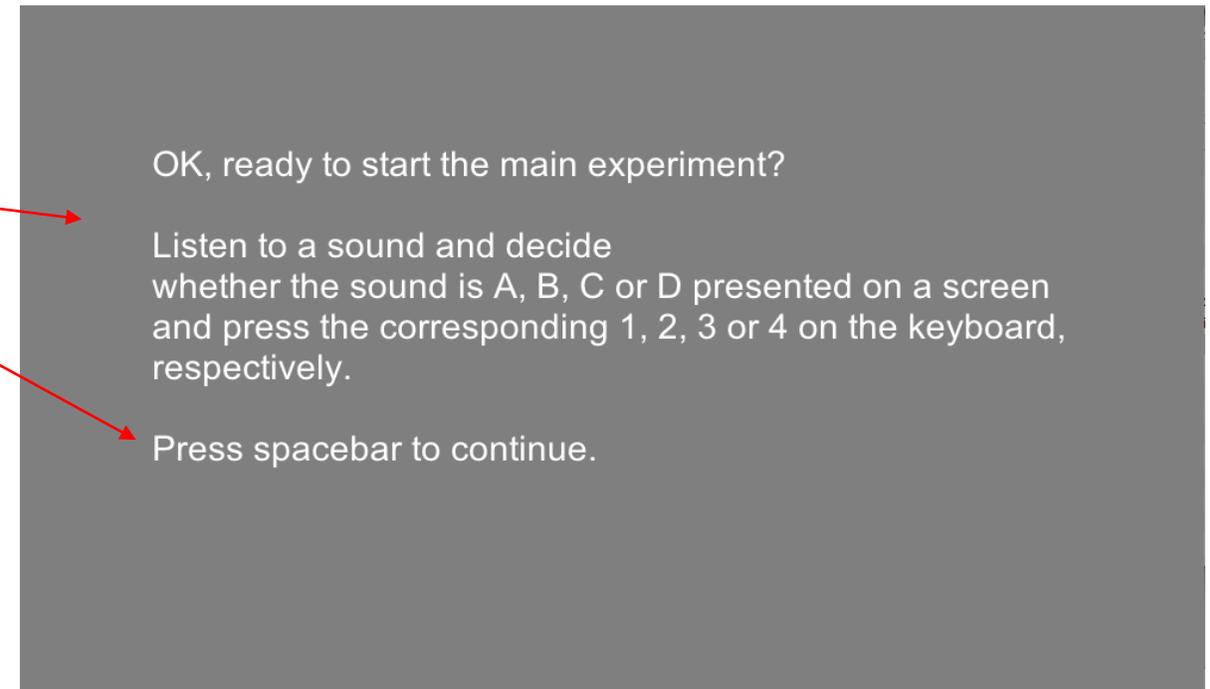
For the Routine [trial1], two Text components, a Sound component and a Keyboard component are needed.

- 4 Click on  in the Component panel and fill out the sound properties (Start time: 0.5, Sound:\$sound, set every repeat)
- 5 Click on  in the Component panel and fill out the keyboard properties (Start time: 0.5, Allowed keys: '1','2','3','4' Store: last key, Store correct: checked, Correct answer:corrAns)

Step5: Add *Components* to the *Routine* [main_instruction]



Screenshot in PsychoPy Builder view

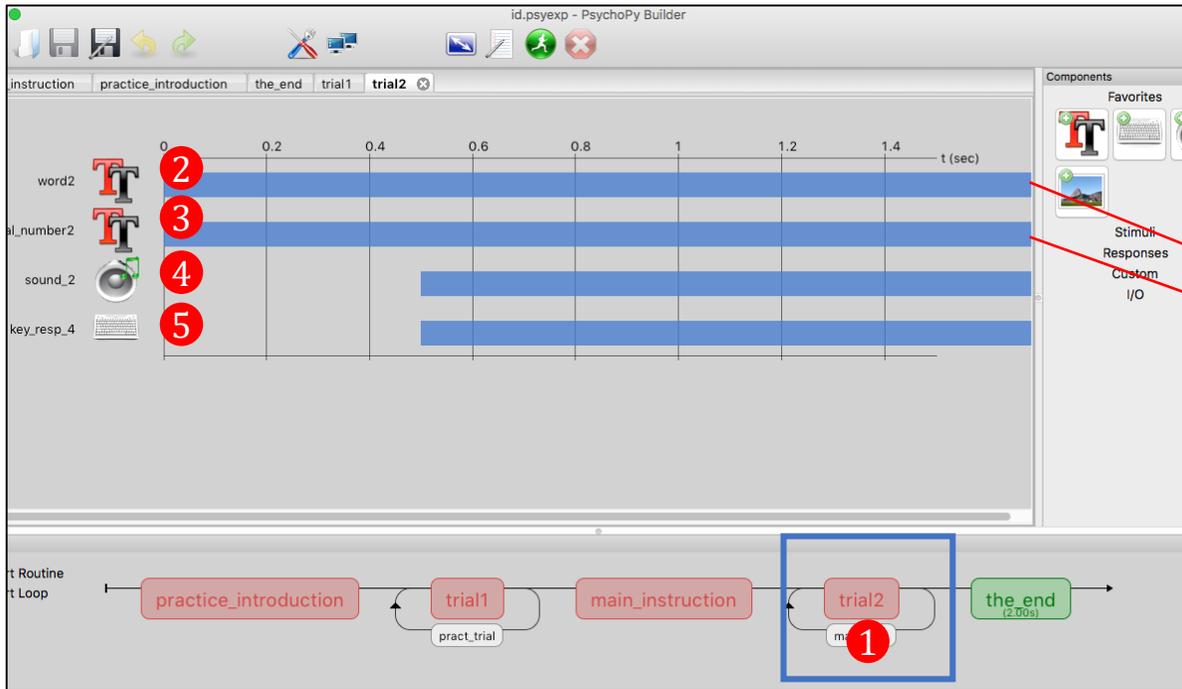


Screenshot in the experiment

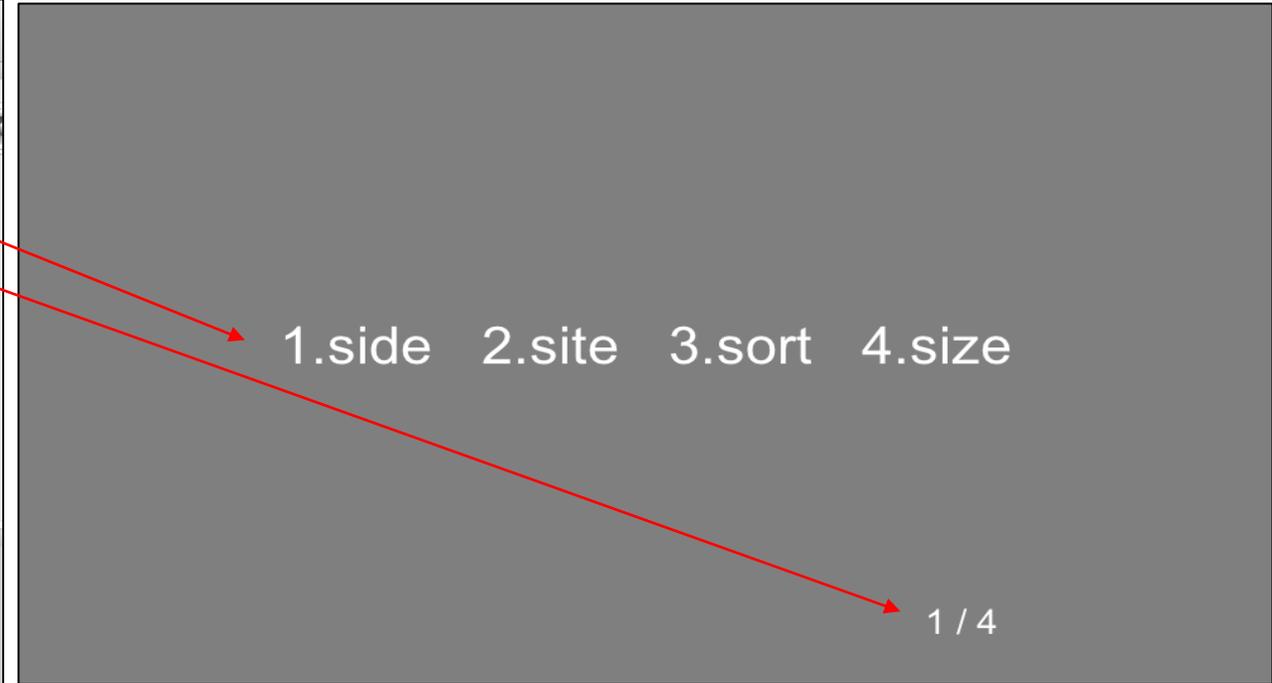
Setting up the *Routine* [main_instruction] is the same as the *Routine* [practice_instruction].

- 1 Select [main_instruction] in the tab, or in the *Flow* panel.
- 2 Add a *Text* component (instruction2) to the *Routine* panel and fill out the text properties.
- 3 Add a *Keyboard* component (key_resp_3) to the *Routine* panel and fill out the keyboard properties.

Step5: Add *Components* to the *Routine* [trial2]



Screenshot in PsychoPy Builder view

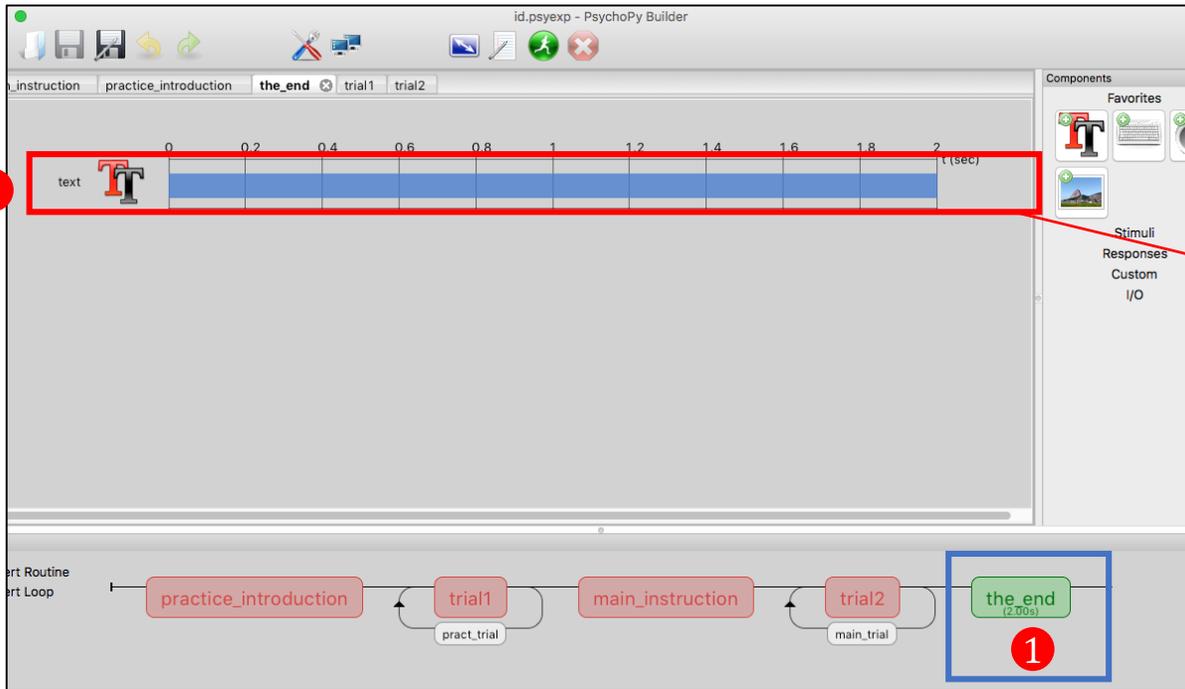


Screenshot in the experiment

Setting up the *Routine* [trial2] is the same as the *Routine* [trial1].

- 1 Select [trial2] in the tab, or in the *Flow* panel.
- 2 3 Add two *Text* components (word2, trial_number2) in the *Routine* panel and fill out the text properties for each.
- 4 6 Add a *Sound* component (sound_2) and a *Keyboard* component (key_resp_4) in the *Routine* panel and fill out the sound properties and keyboard properties.

Step5: Add *Components* to the *Routine* [the_end]



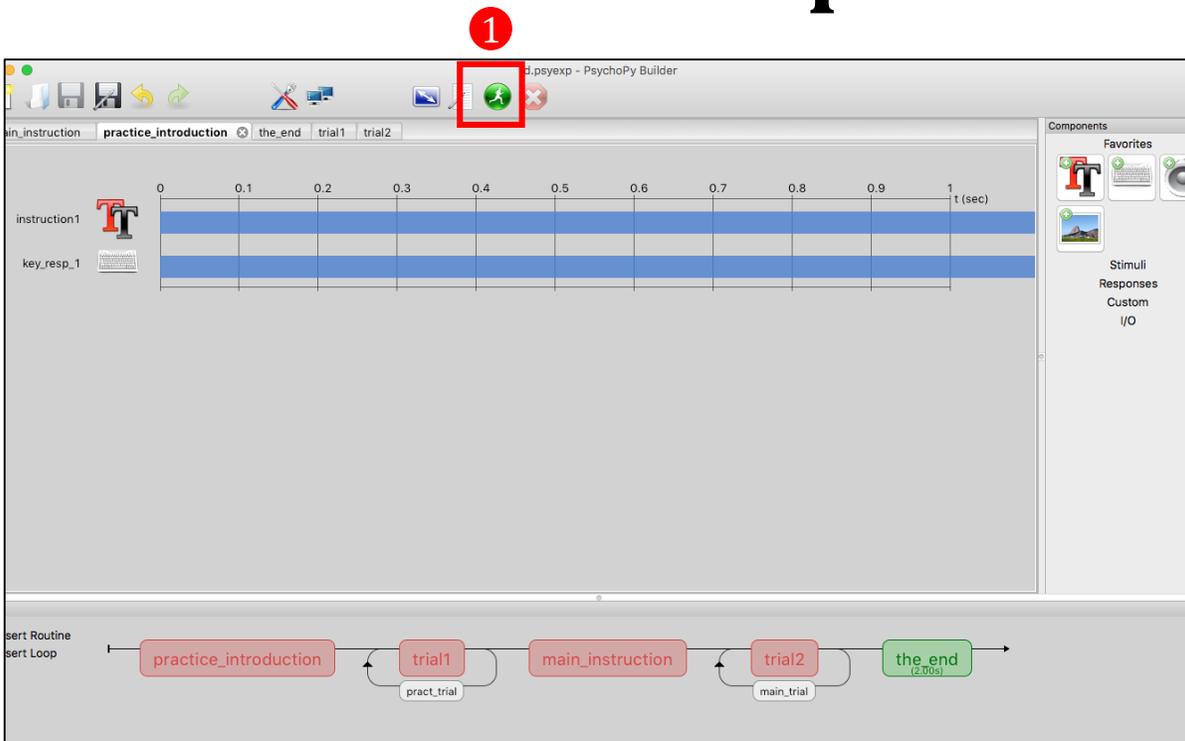
Screenshot in PsychoPy Builder view



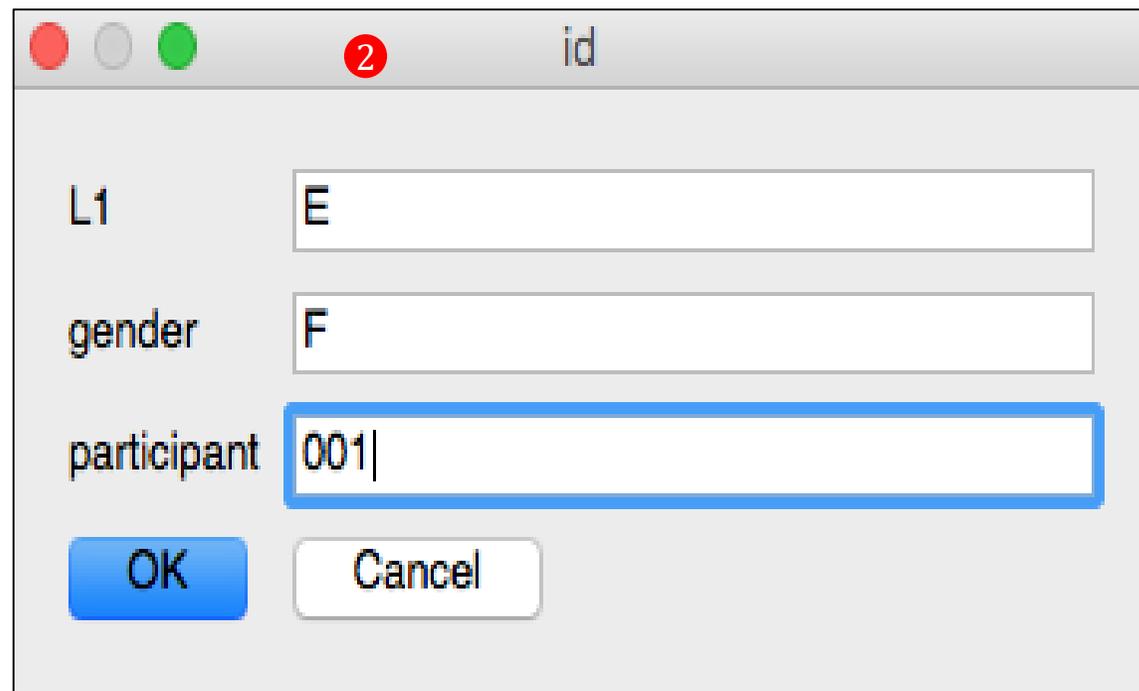
Screenshot in the experiment

- 1 Select the *Routine* [the_end] in the tab, or in the *Flow* panel.
- 2 Click or  In the Component panel and fill out the text properties: Stop duration:2, Letter height:\$0.1 Position\$(0,0), Wrap width\$ 2

Step6: Run an experiment



Screenshot in PsychoPy Builder view



Screenshot in the experiment

- 1 To run an experiment, either press **the green button** with the running man icon or press Ctrl (Command) + R
- 2 Fill out the pop-up window for the participant information.

Analyze your data

- PsychoPy saves several data files for different uses: a Microsoft Excel (spreadsheet) file, a ‘psydat’ file, and a ‘log’ file.
- To find these
 - Go to the folder where you saved the experiment.
 - There will be a new folder inside that (next to the psyexp file) called “data”.
 - Inside the data folder will be a Microsoft Excel file named by your username and the date.
- E.g. Output data file from the identification task in which a participant (Gender: F, L1: English) identify a sound among four sounds.

sound	w4	w3	w2	w1	corrAns	pract_trial.t	pract_trial.t	pract_trial.t	pract_trial.t	main_trial.t	main_trial.t	main_trial.t	main_trial.t	key_resp_2	key_resp_2	key_resp_2	key_resp_4	key_resp_4	key_resp_4	participant	gender	frameRate	expName	L1	date
sound/ride.v	write	right	ride	light	2	0	0	0	0					1	0	19.32195				1	F	59.89108	id	E	2017_Oct_18_1923
sound/top.v	pop	cop	top	stop	2	0	1	1	1					2	0	18.41558				1	F	59.89108	id	E	2017_Oct_18_1923
sound/side.v	size	sort	site	side	1					0	0	0	0						2	0	34.47897	id	E	2017_Oct_18_1923	
sound/tight.v	think	though	tight	tide	2					0	1	1	1						2	0	0.203089	id	E	2017_Oct_18_1923	
sound/ride.v	write	right	ride	light	2					0	2	2	2						2	0	0.069742	id	E	2017_Oct_18_1923	
sound/top.v	pop	cop	top	stop	2					0	3	3	3						2	0	0.103932	id	E	2017_Oct_18_1923	

References & Useful information

- **Na-Young Ryu. 2017.** Psychopy tutorials for common linguistic experiments.
<http://individual.utoronto.ca/rrrnny/experiments.html>
- **Peirce, JW (2007).** PsychoPy - Psychophysics software in Python. *J Neurosci Methods*, 162(1-2):8-13
- **PsychoPy:**
<http://www.psychopy.org>
- **RatingScale & RatingScale Component**
<http://www.psychopy.org/api/visual/ratingscale.html>
<http://www.psychopy.org/builder/components/ratingscale.html>
- **YouTube tutorials:**
<https://www.youtube.com/watch?v=VV6qhuQgsil>
<https://www.youtube.com/watch?v=WKJBbVnQkjo>
- **Sound issues:** When PsychoPy does not work properly due to the sound device, go to PsychoPy Preferences > General > audioDevice (built-in Output).

