

Building linguistic experiments in PsychoPy: Production task

Department of Linguistics

University of Toronto

Oct 2017

Na-Young Ryu

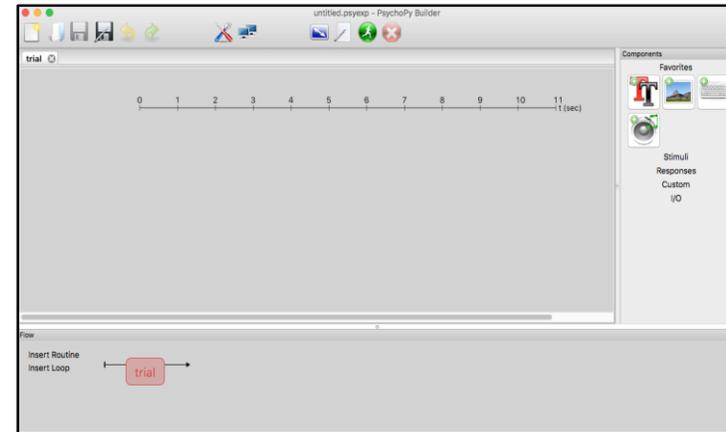
Building an experiment is like a building a house!

- 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

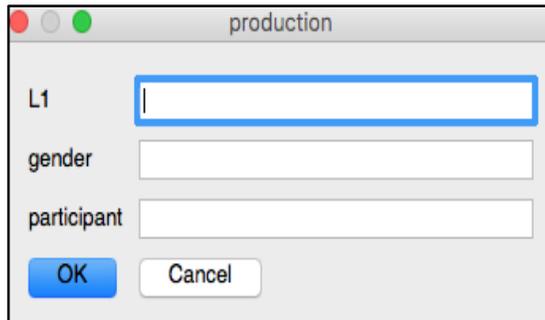
Download our materials for your experiments

- **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>

Production Task: Goals & Procedures

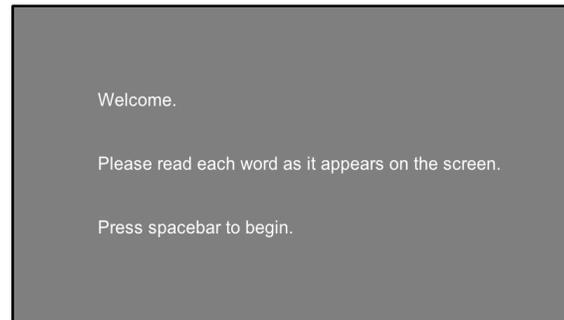
- This production experiment consists of five routines: **instruction 1** followed by **trial1** for the practice session and then **Instruction2** followed by **trial2** for the main session and the “**thank you**” message to participants at the end.
- In our tutorial, we provide two versions of the production experiments:
 - (1) ‘**Auto**’ version – each word appears every 3 seconds so that no ticking noise from the keyboard is inserted while recording.
 - (2) ‘**Keyboard response**’ version – participants have to press the spacebar to progress to the next word after they read a word.

1. Filling out participant information



A screenshot of a dialog box titled "production". It contains three input fields: "L1" (with a blue border), "gender", and "participant". At the bottom, there are "OK" and "Cancel" buttons.

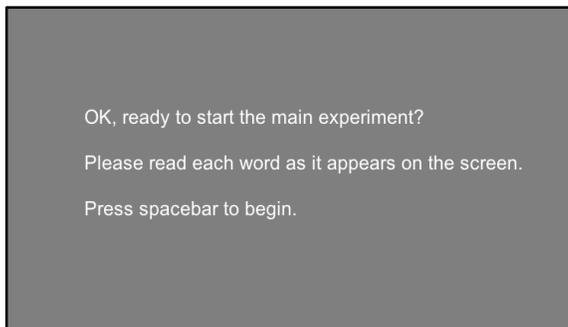
2. Showing instruction 1



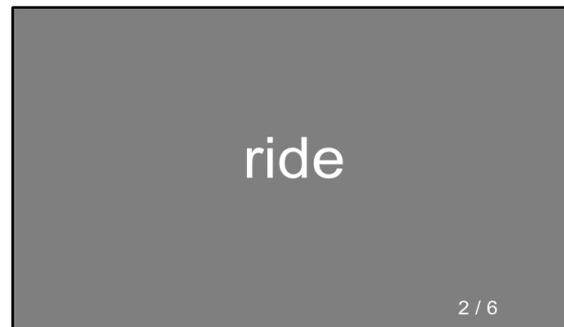
3. Displaying stimuli



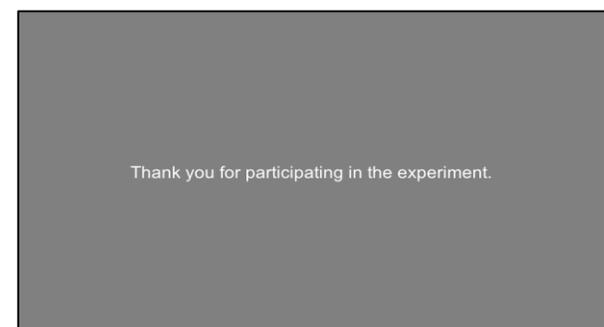
4. Showing instruction 2



5. Displaying stimuli



6. “Thank you” message



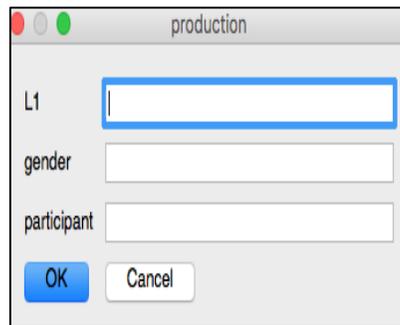
How to build the production experiment?

- **Step1:** Create a folder to store your experiment files
 - We have already provided two Excel files containing word lists for the production experiment.
 - Please find them in the folder: “1-1.production(auto)” and “1-2.Production(key_response)”.
- **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: Wordlist

	A
1	word
2	lied
3	light
4	ride

Step 2: Participant info



production

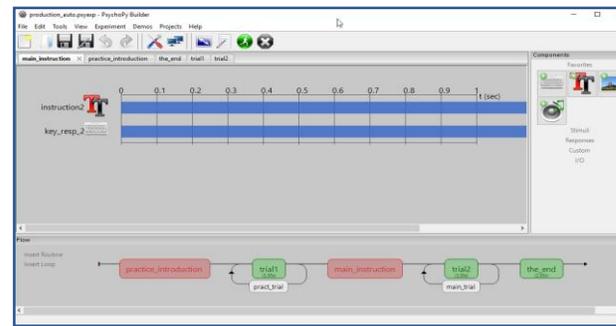
L1

gender

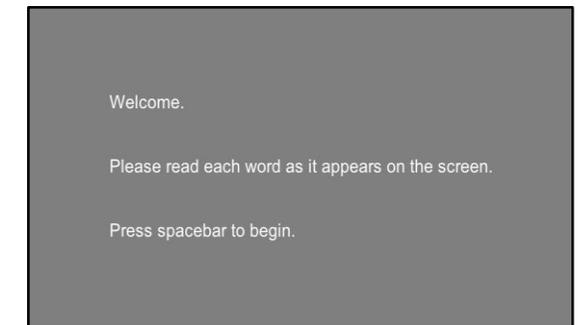
participant

OK Cancel

Step 3-5: Create *Routines* and *loops*



Step 6 : Run an experiment



PsychoPy *Builder* interface

The screenshot displays the PsychoPy Builder interface for a file named 'production_auto.psyexp'. The interface is divided into three main panels:

- 1 The Routine panel:** Located at the top, it shows a timeline from 0 to 1 second. Two components are visible: 'instruction2' (represented by a large red 'T' icon) and 'key_resp_2' (represented by a keyboard icon). Both components are active throughout the 1-second duration, indicated by blue horizontal bars.
- 2 The Flow panel:** Located at the bottom, it shows a sequence of routines: 'practice_introduction' (red box), 'trial1 (3.00s)' (green box) with a sub-routine 'pract_trial' (white box), 'main_instruction' (red box), 'trial2 (3.00s)' (green box) with a sub-routine 'main_trial' (white box), and 'the_end (2.00s)' (green box). Arrows indicate the flow between these routines.
- 3 The Component panel:** Located on the right side, it contains a 'Components' list with 'Favorites' (a keyboard icon, a large red 'T' icon, and a landscape image icon) and 'Stimuli' (a camera icon). Below this are categories for 'Responses', 'Custom', and 'I/O'.

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

The *Routine* Panel

The image shows the PsychoPy Builder interface for a routine named 'rating.psyexp'. The main window displays a timeline from 0 to 1 second. Two routines are visible: 'instruction2' and '_resp_2'. The 'instruction2' routine is highlighted with a red box and contains a text component (represented by a large 'T' icon). The '_resp_2' routine is also highlighted with a red box and contains a keyboard component (represented by a keyboard icon). To the right of the timeline is the 'Components' panel, which shows a list of components available for the routine. A red box labeled '1' highlights the tabs at the top of the Components panel, which are labeled 'practice_instruction', 'the_end', 'trial1', and 'trial2'. A red arrow points from the text 'These tabs show Routines.' to these tabs. Another red box labeled '2' highlights the 'Text component' and 'Keyboard component' in the Components panel. Red arrows point from the text 'Text component' and 'Keyboard component' to these respective components. Below the timeline is a flowchart showing the sequence of routines: 'practice_instruction' -> 'trial1' (with a sub-routine 'pract_trial') -> 'main_instruction' -> 'trial2' (with a sub-routine 'main_trial') -> 'the_end'. The text 'The Routine panel' is written in red in the center of the main window.

1 Components

These tabs show Routines.

2

Text component

Keyboard component

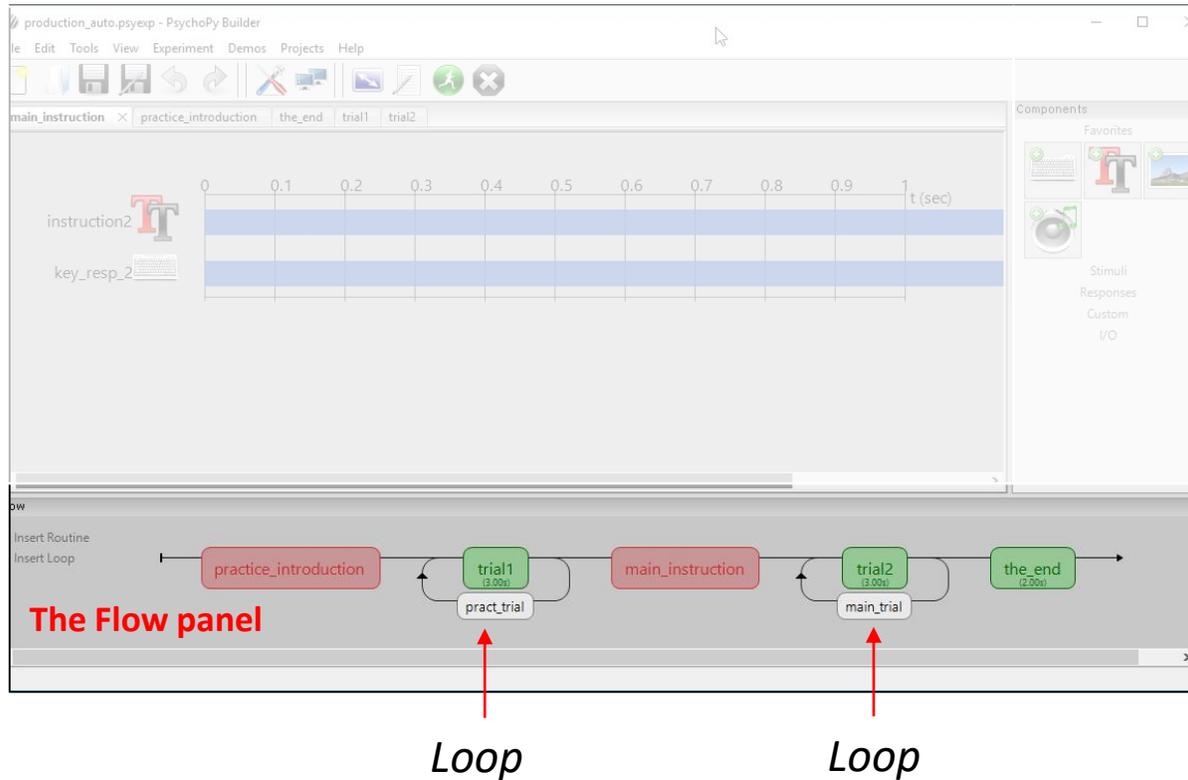
The Routine panel

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

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

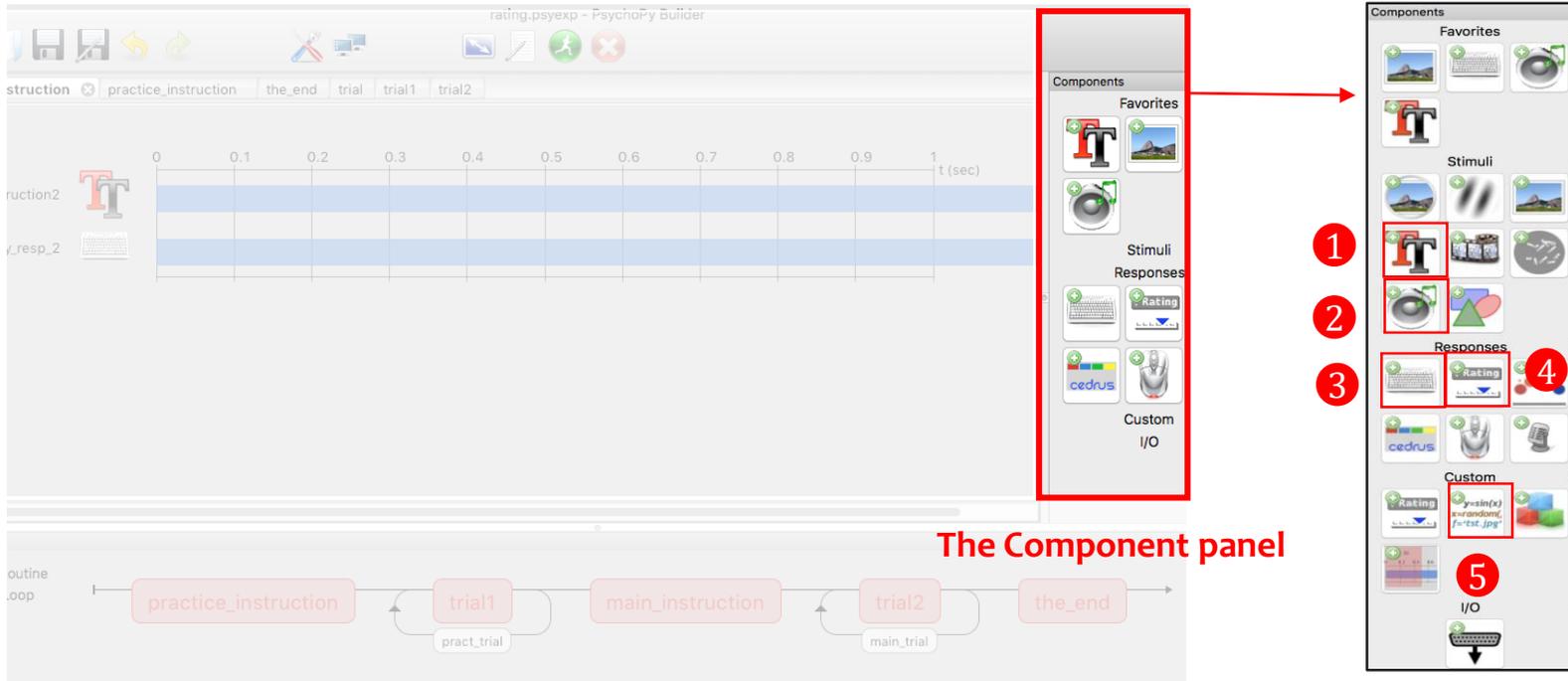
2. The Routine shown here has a **Text** and **Keyboard component**.

PsychoPy 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:

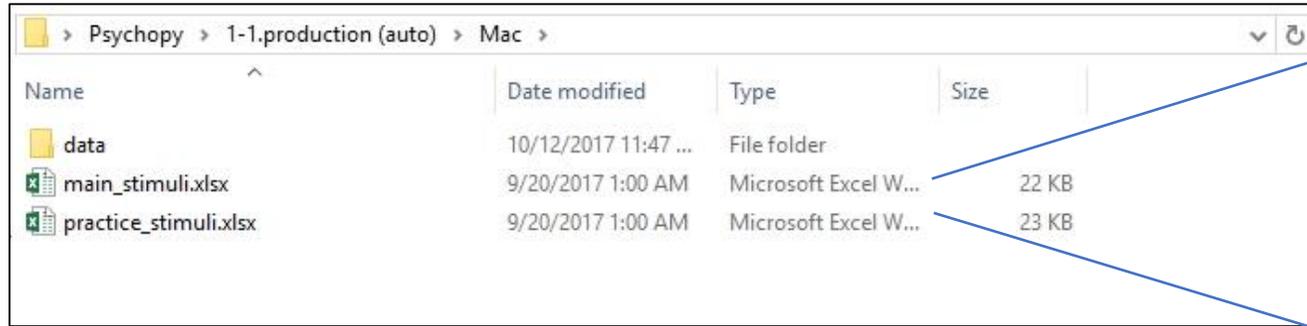
1. **Text Component** - Display text on the screen.
2. **Sound Component** - Play sounds.
3. **Keyboard Component** - Receive input from the keyboard.
4. **RatingScale Component** - Collect a numeric rating or a choice from a few alternatives, via the mouse, the keyboard or both.
5. **Code Component**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

Names for the production experiment

- **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	
Keyboard component	key_resp_1		key_resp_2		

Step1: Create Excel files with your word list.



main_stimuli.xlsx

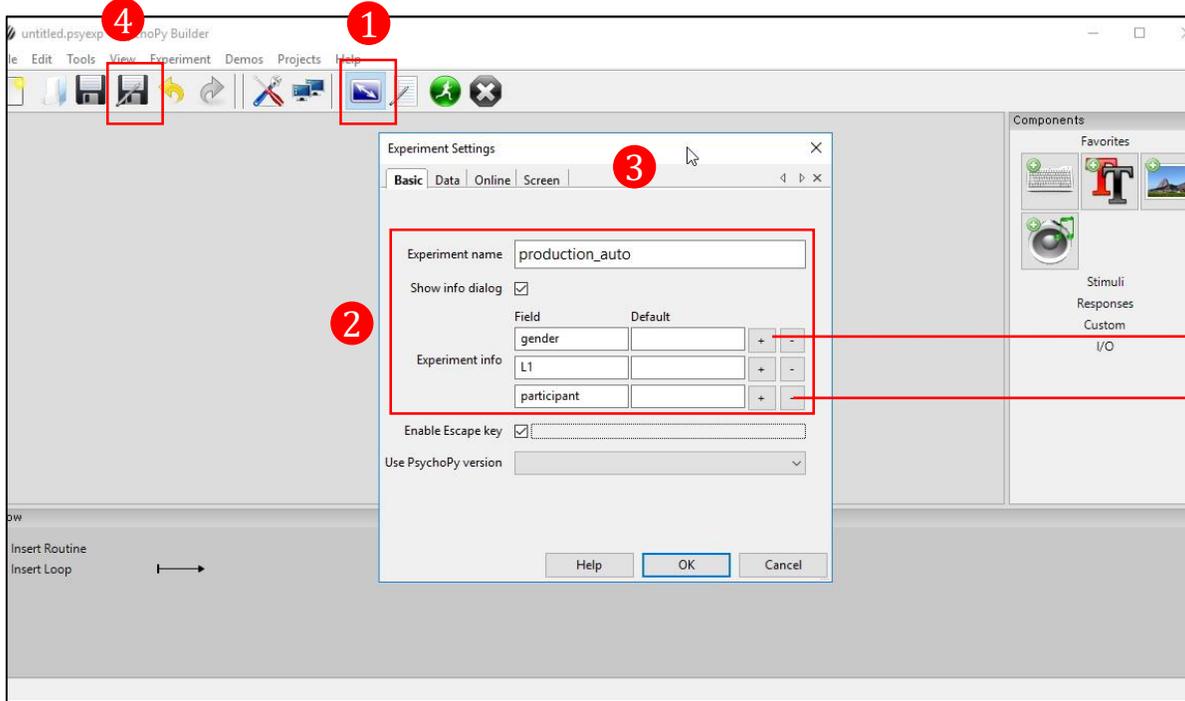
	A
1	word
2	lied
3	light
4	ride

practice_stimuli.xlsx

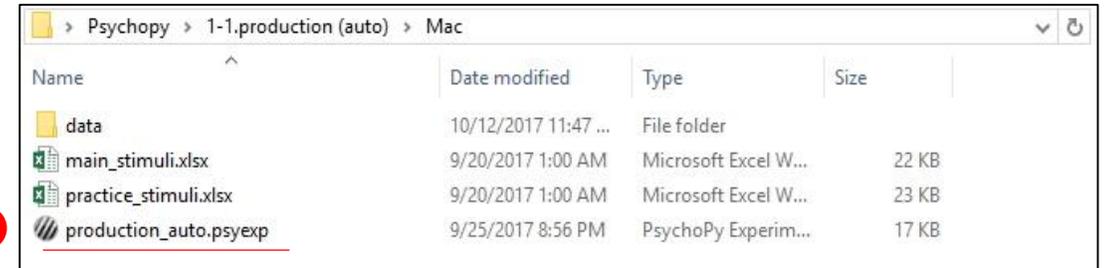
	A
1	word
2	rider
3	writer
4	louder

- Create a folder named “production(auto)” on your computer to store experiment files.
- In the production task, we need two Excel files with stimuli for the practice and main trials (‘main_stimuli.xlsx’ and ‘practice_stimuli.xlsx’)
- We also need to create a folder named “data” in which an output file of this experiment will be automatically generated.

Step2: Alter experiment settings

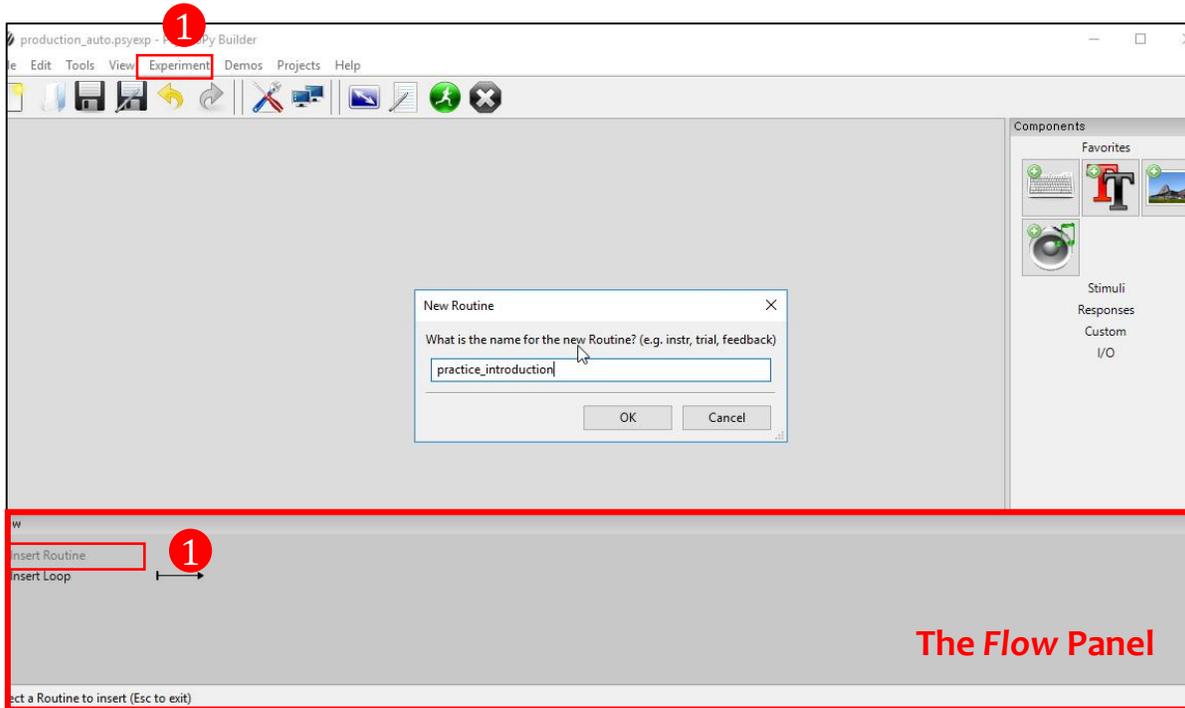


-  To add experiment info
-  To delete experiment info



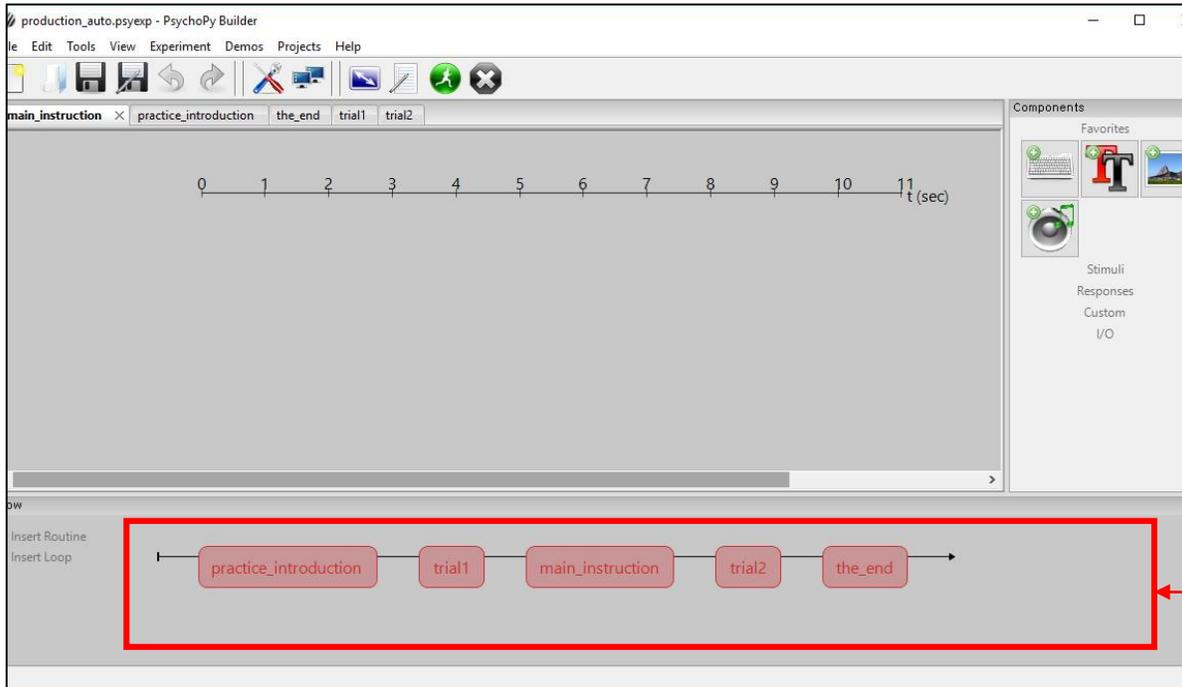
- 1 Click on  on the top of the screen for **experiment settings**.
- 2 Write an experiment name, e.g. “production_auto”, and modify the experiment information (e.g. participant ID, gender, L1) by clicking +/- icons.
- 3 If you would like to change the forms of output files or the size of the window,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

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*.

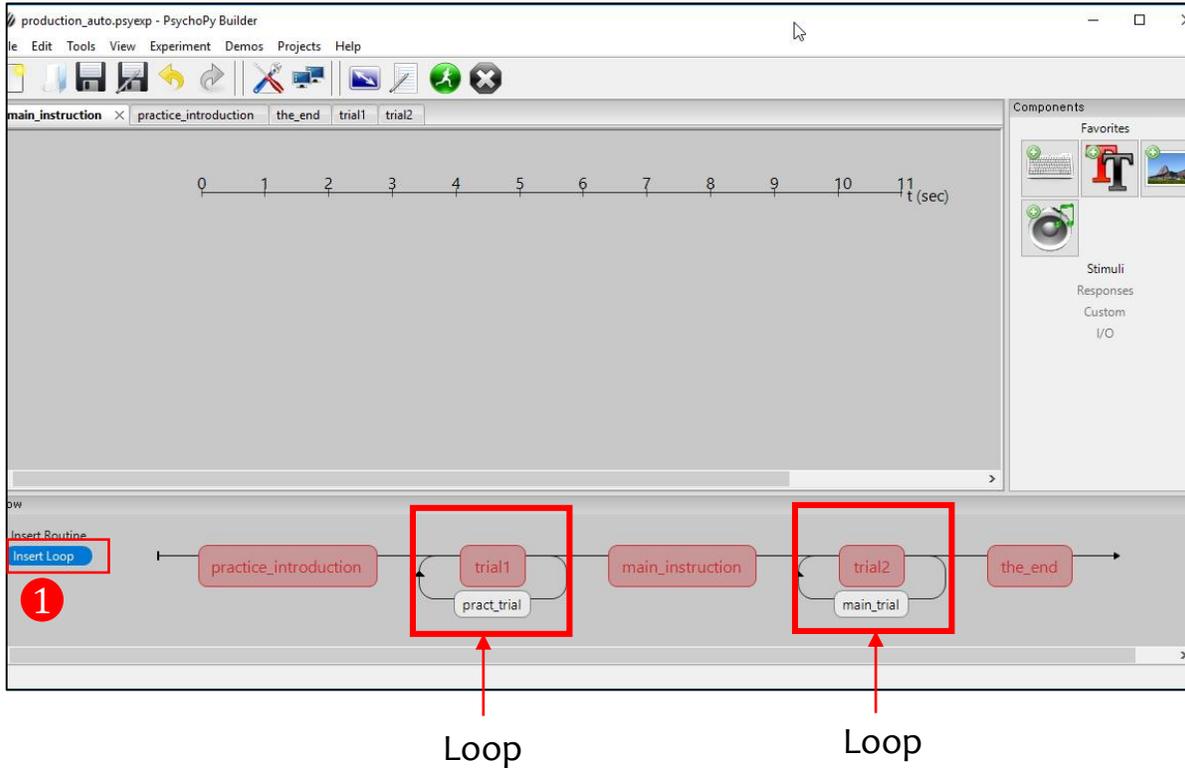
Step4: Add the *Routines* to the *Flow*



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 production 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*



The 'Loop Properties' dialog box is shown. It has a 'Name' field with 'pract_trial', a 'loopType' dropdown set to 'random', and a checked 'Is trials' checkbox. Below these are fields for 'random seed \$', 'nReps \$' (set to 1), and 'Selected rows \$'. The 'Conditions' field contains 'practice_stimuli.xlsx' and a 'Browse...' button. Below the conditions, it says '3 conditions, with 1 parameters [word]'. A text box below the dialog says: 'When inserting a loop into the *Flow* you can browse to find the file you wish to use for this.' To the right of the dialog, a list of loop types is shown: 'random' (checked), 'sequential', 'fullRandom', 'staircase', and 'interleaved staircases'. A red box highlights the 'nReps \$' and 'Selected rows \$' fields, with an arrow pointing to the text 'No of the repetition'. A table below the dialog shows a list of words: 'word', 'lied', 'light', 'ride'. A red arrow points from the 'Browse...' button to this table. Green arrows point from the text '1 parameter' and '3 conditions' to the 'word' and 'lied' rows respectively.

	A
1	word
2	lied
3	light
4	ride

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*

instruction1 Properties

Basic | Advanced

Name: instruction1

Start: time (s) | 0.0 | Expected start (s)

Stop: duration (s) | | Expected duration (s)

Color: white | constant

Font: Arial | constant

Letter height \$: 0, 1 | constant

Position [x,y] \$: (0, 0) | constant

Text: welcome.
Please read each word as it appears on the screen.
Press spacebar to begin.

constant

Help OK Cancel

- 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]

production_auto.psyexp - PsychBuilder

File Edit Tools View Experiment Demos Projects Help

main_instruction practice_introduction the_end trial1 trial2

Components

Favorites

Stimuli Responses Custom I/O

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 t (sec)

instruction1

key_resp_1

practice_introduction trial1 main_instruction trial2 the_end

1

2

3

Instruction1 Properties

Basic Advanced

Name: instruction1

Start time (s): 0.0 Expected start (s):

Stop duration (s): Expected duration (s):

Color: White constant

Font: Arial constant

Letter height \$: 0.1 constant

Position [x,y] \$: [0, 0] constant

Text: welcome. Please read each word as it appears on the screen. Press spacebar to begin.

Help OK Cancel

Clear the duration

Keyboard Properties

Basic

Name: key_resp_1

Start time (s): 0.0 Expected start (s):

Stop duration (s): Expected duration (s):

Force end of Routine:

Allowed keys \$: 'space' constant

Store: nothing

Store correct:

Discard previous:

sync RT with screen:

Help OK Cancel

2 Text properties

3 Keyboard properties

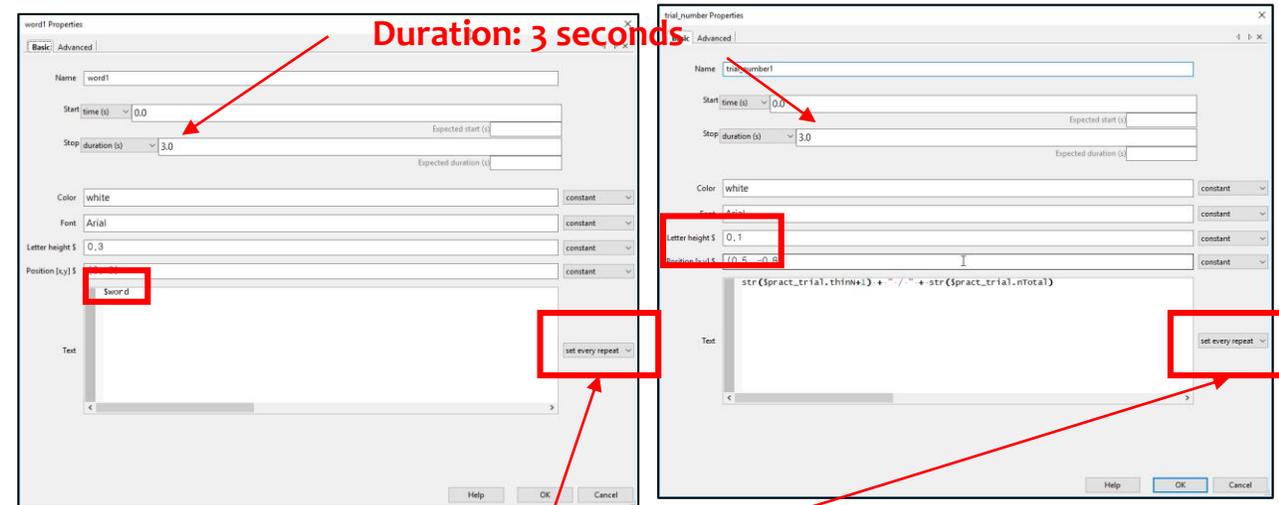
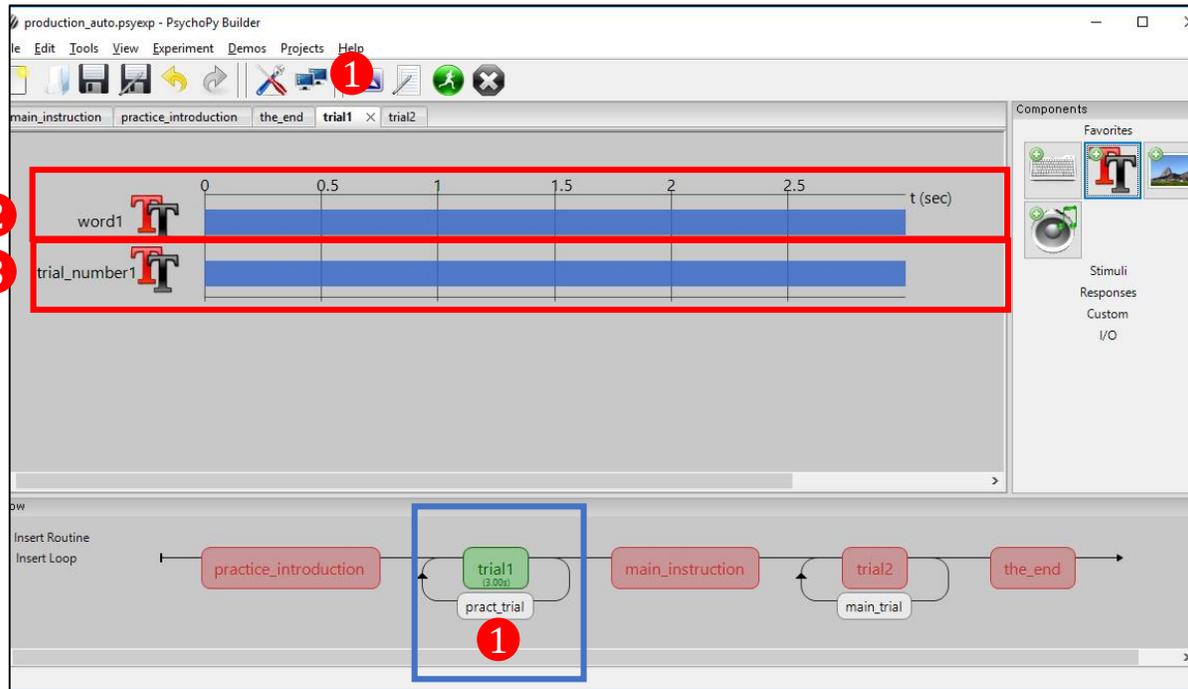
For the *Routine* [practice_instruction], a *Text* component and a *Keyboard* component are needed.

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)

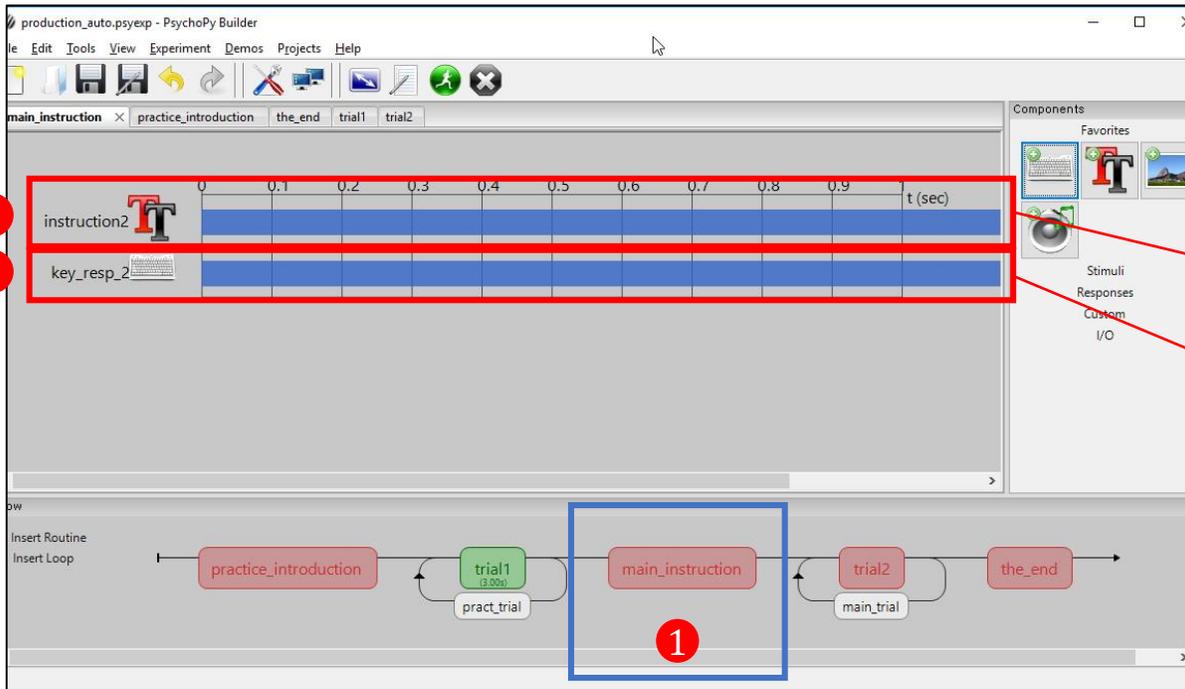
'set every repeat'

It will be updated every repeat of the Routine.

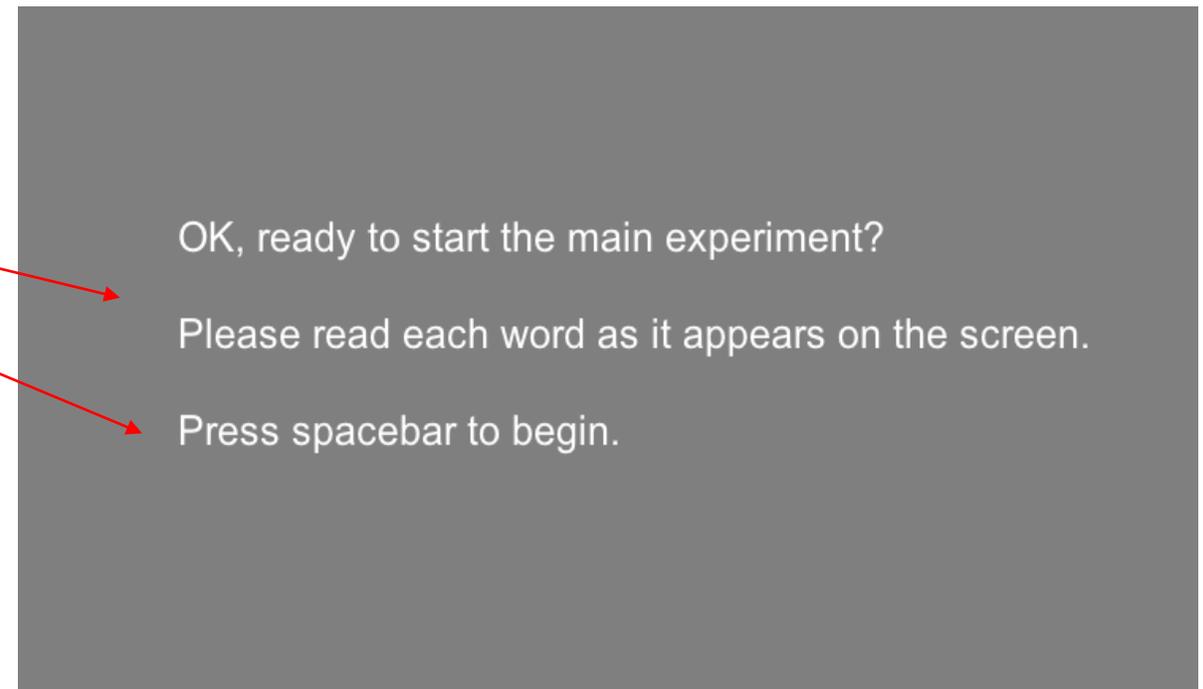
For Routine [trial1], two Text components are needed.

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: 3**, Font: Arial, Letter height: 0.3, Position: (0.0, 0.3), Text: **\$word** (*the \$ symbol announces that this is not text)
3. Click on  in the Component panel and fill out the text properties: **Stop duration: 3**, Font: Arial, Letter height: 0.1, Position: (0.5, -0.8), Text: **str(\$pract_trial.thisN+1) + " / " + str(\$pract_trial.nTotal)**, set every repeat

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



Screenshot in PsychoPy Builder view

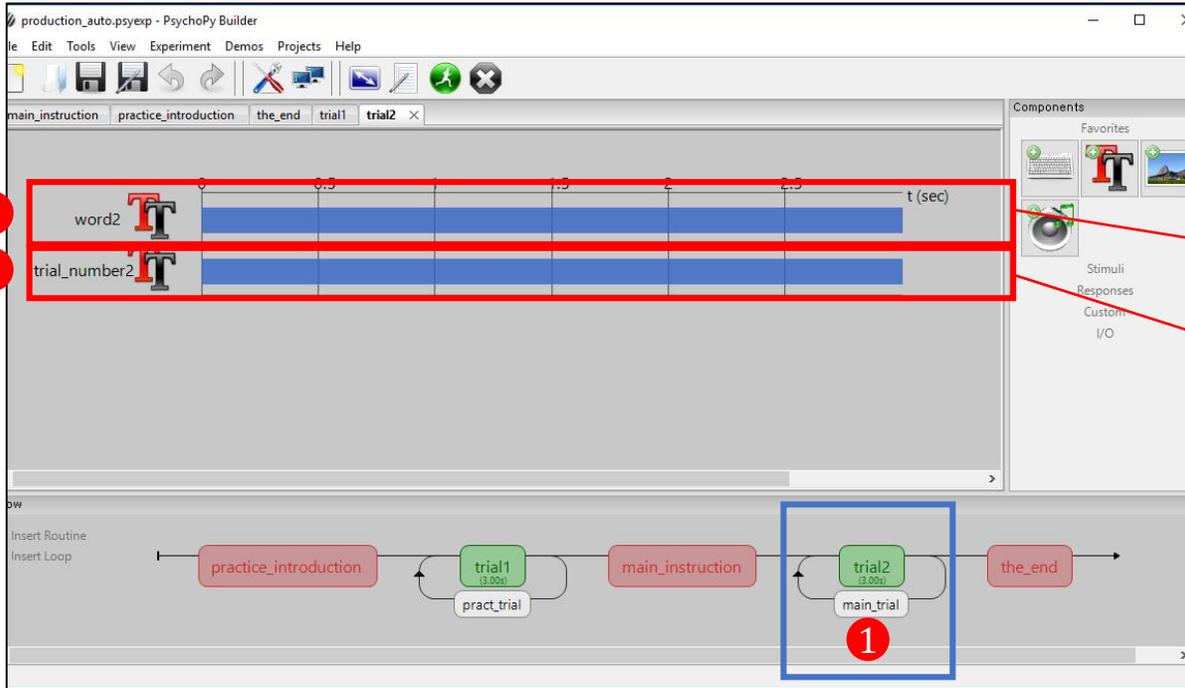


Screenshot in the experiment

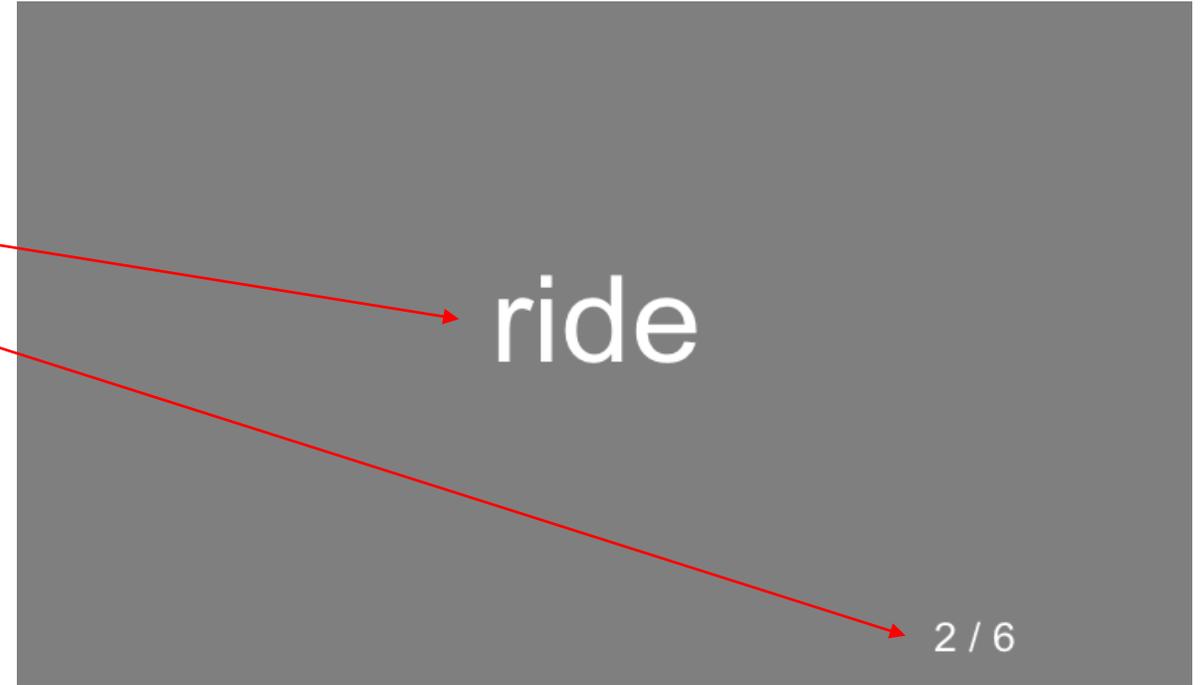
Setting up the *Routine* [main_instruction] is the same as the *Routine* [practice_instruction].

- 1 Select [main_instruction] in the tab, or in the *Flow*.
- 2 Add a *Text* component (instruction2) to the *Routine* panel and fill out the text properties.
- 3 Add a *Keyboard* component (key_resp_2) to the *Routine* panel and fill out the keyboard properties.

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



Screenshot in PsychoPy Builder view

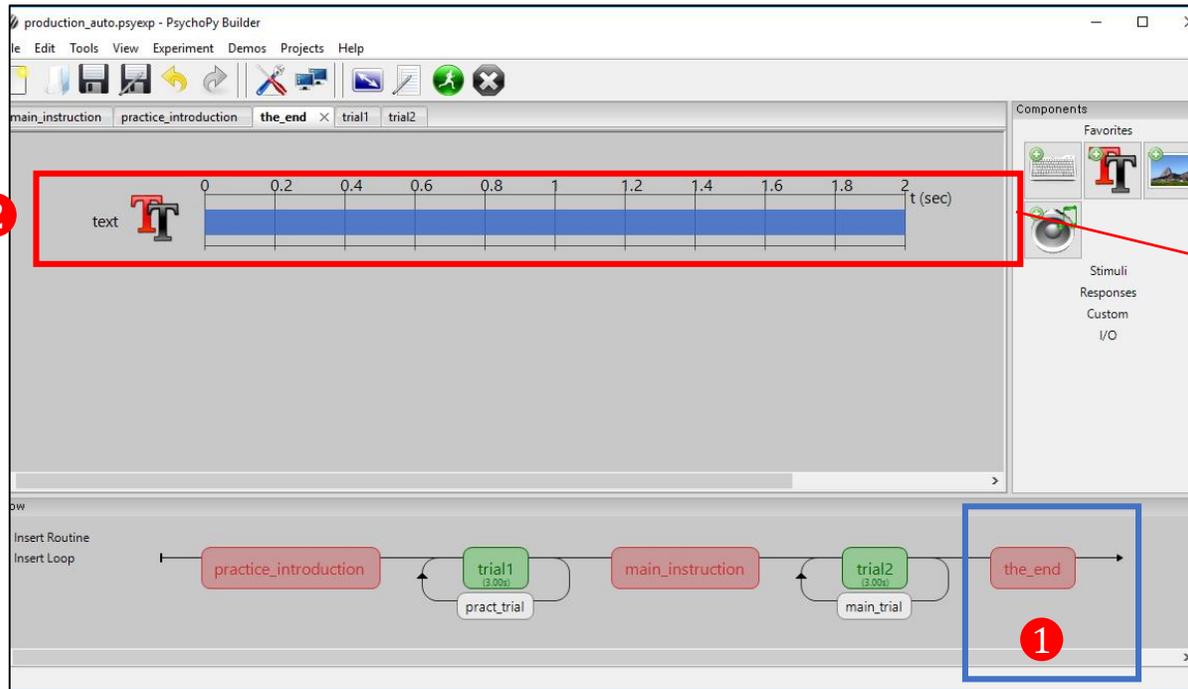


Screenshot in the experiment

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

- 1 Select [trial2] in the tab, or in the *Flow* panel.
- 2 Add a *Text* component (word2) in the *Routine* panel and fill out the text properties.
- 3 Add a *Text* component(trial_number2) and fill out the text properties Text: `str($main_trial.thisN+1) + “ / ” +str($main_trial.nTotal)`.

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



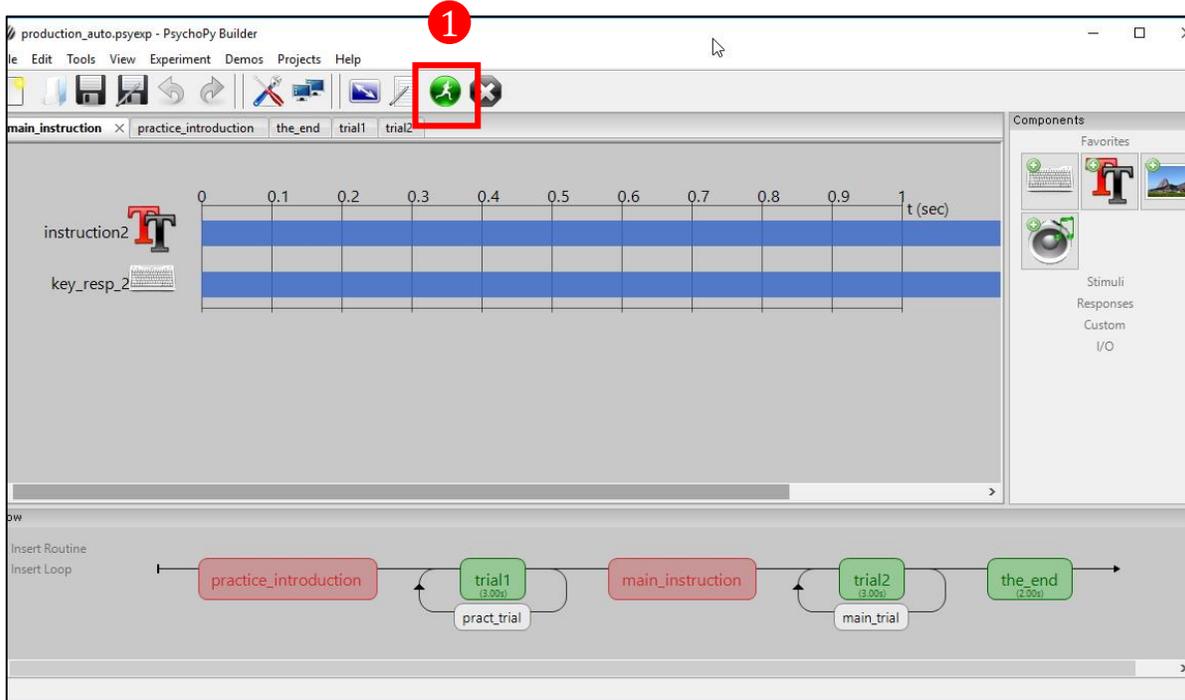
Screenshot in PsychoPy Builder view



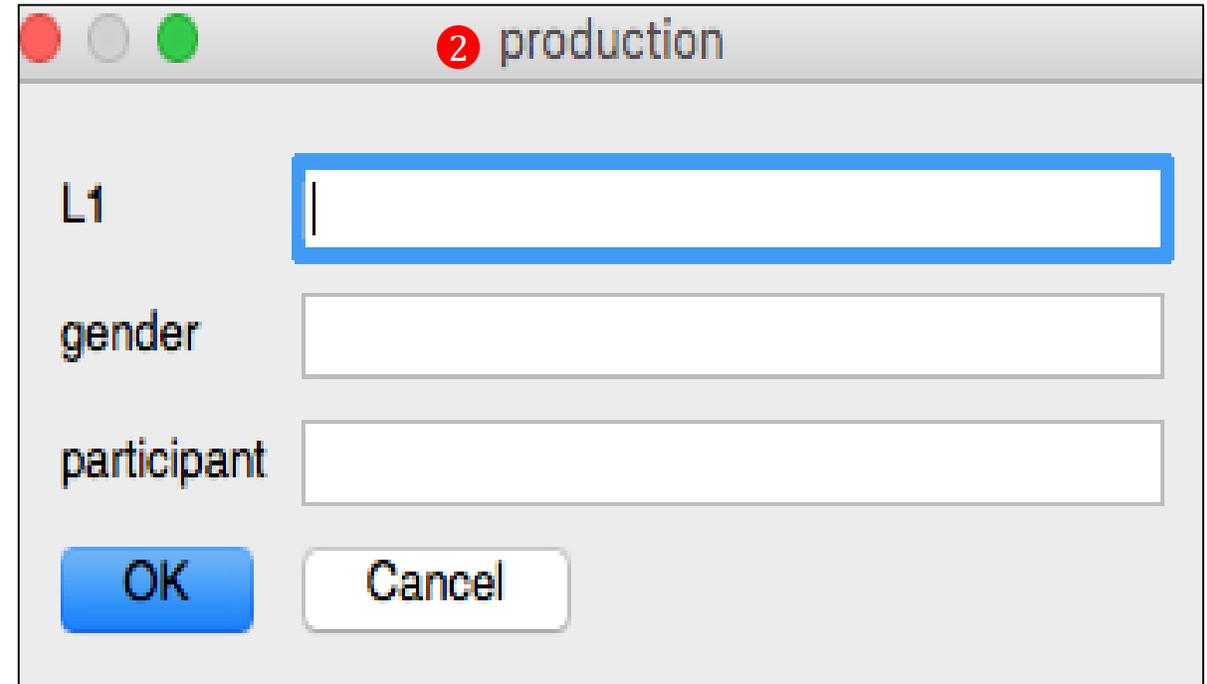
Screenshot in the experiment

- 1 Select the *Routine* [the_end] in the tab, or in the *Flow* panel.
- 2 Click on  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 and start running the rating experiment.

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 an AX discrimination task in which participants judged whether two sounds are the same or different.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
sound1	sound2	corrAns	pract_trial.th	pract_trial.th	pract_trial.th	pract_trial.th	main_trial.th	main_trial.th	main_trial.th	main_trial.th	key_resp_2.l	key_resp_2.c	key_resp_2.r	key_resp_4.l	key_resp_4.c	key_resp_4.r	participant	gender	frameRate	expName	L1	date	
sound/A.wav	sound/A.wav	1	0	0	0	0					1	1	2.4849689				1	m	59.2270261	AX	m	2017_Oct_14_1904	
sound/l.wav	sound/A.wav	0	0	1	1	1					0	1	1.86030602				1	m	59.2270261	AX	m	2017_Oct_14_1904	
sound/l.wav	sound/l.wav	1					0	0	0	1				1	1	1.25246692	1	m	59.2270261	AX	m	2017_Oct_14_1904	
sound/O.wav	sound/A.wav	0					0	1	1	0				0	1	1.384902	1	m	59.2270261	AX	m	2017_Oct_14_1904	
sound/l.wav	sound/l.wav	1					1	0	2	1				1	1	0.62529302	1	m	59.2270261	AX	m	2017_Oct_14_1904	
sound/O.wav	sound/A.wav	0					1	1	3	0				0	1	0.7119751	1	m	59.2270261	AX	m	2017_Oct_14_1904	

References & Useful websites

- **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>
- **Flow**
<http://www.psychopy.org/builder/flow.html>
- **Text Component**
<http://www.psychopy.org/builder/components/text.html>
- **Keyboard Component**
<http://www.psychopy.org/builder/components/keyboard.html>
- **YouTube tutorials**
<https://www.youtube.com/watch?v=VV6qhuQgsil>
<https://www.youtube.com/watch?v=WKJBbVnQkjo>