Building linguistic experiments in PsychoPy: AX discrimination task

Department of Linguistics
University of Toronto
Oct 2017

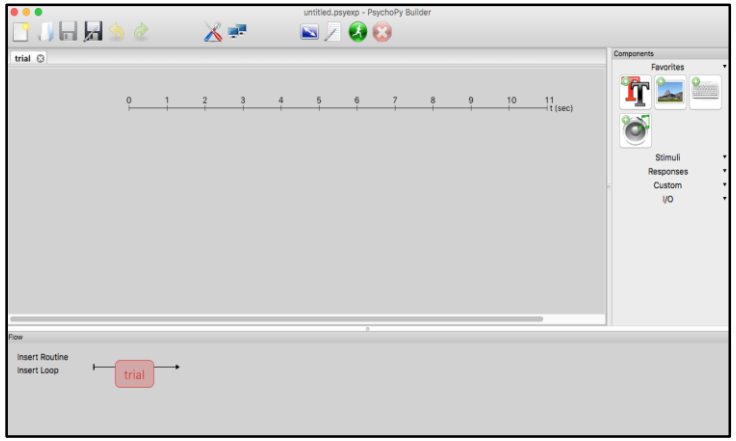
Na-Young Ryu

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

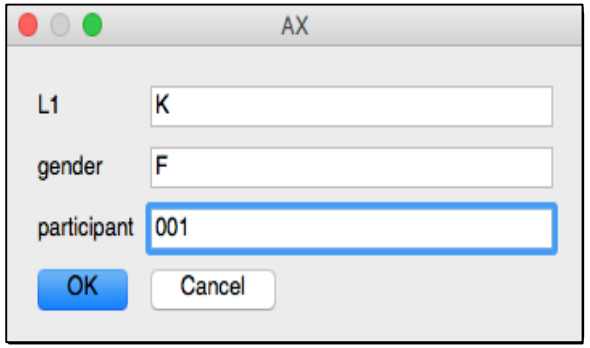
- **Why build experiments using the PsychoPy builder?**
  - It is free and easy to generate a wide range of linguistic experiments in the *Builder* view.
  - The builder view allows you to create experiments visually, so the Python programming language is not needed.
  - Output data is automatically 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**

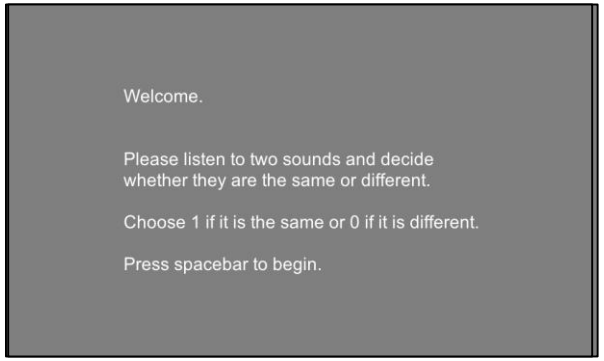
AX discrimination Task: Goals & Procedures

- **Goals:** We provide instructions on how to create an AX discrimination experiment using the PsychoPy Builder view (Peirce 2007) . The AX discrimination task is designed to ask participants to listen to two sounds and decide whether they are the same or different .
- **Procedures:** Show **Instruction 1** followed by **trial1** 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

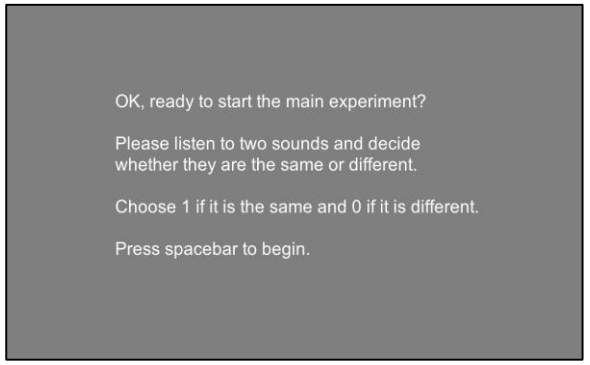


2. Showing Instruction 1



3. Displaying stimuli and getting responses

4. Showing Instruction 2



5. Displaying stimuli and getting responses

6. Displaying "Thank you" message

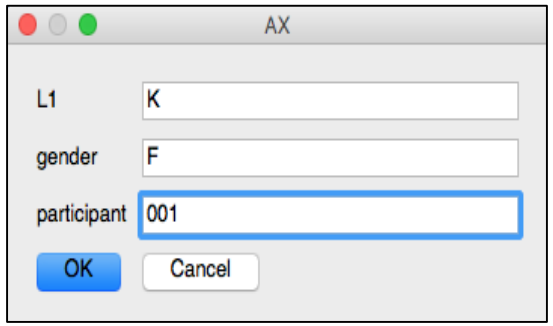
How to build the identification task?

- **Step1**: Create an Excel file with your name of sound 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.

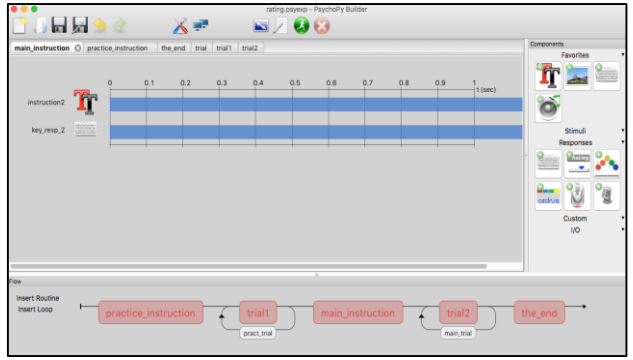
Step 1: Excel file with sound files

sound1	sound2	corrAns
sound/O.wav	sound/A.wav	0
sound/I.wav	sound/I.wav	1

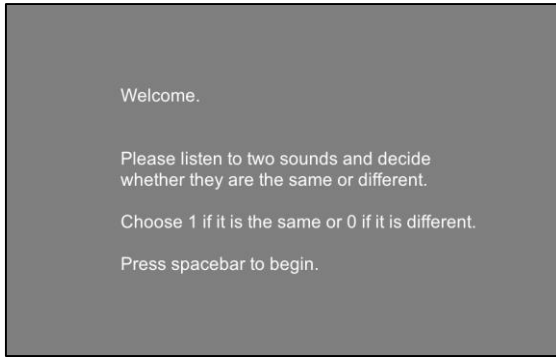
Step 2: Participant info



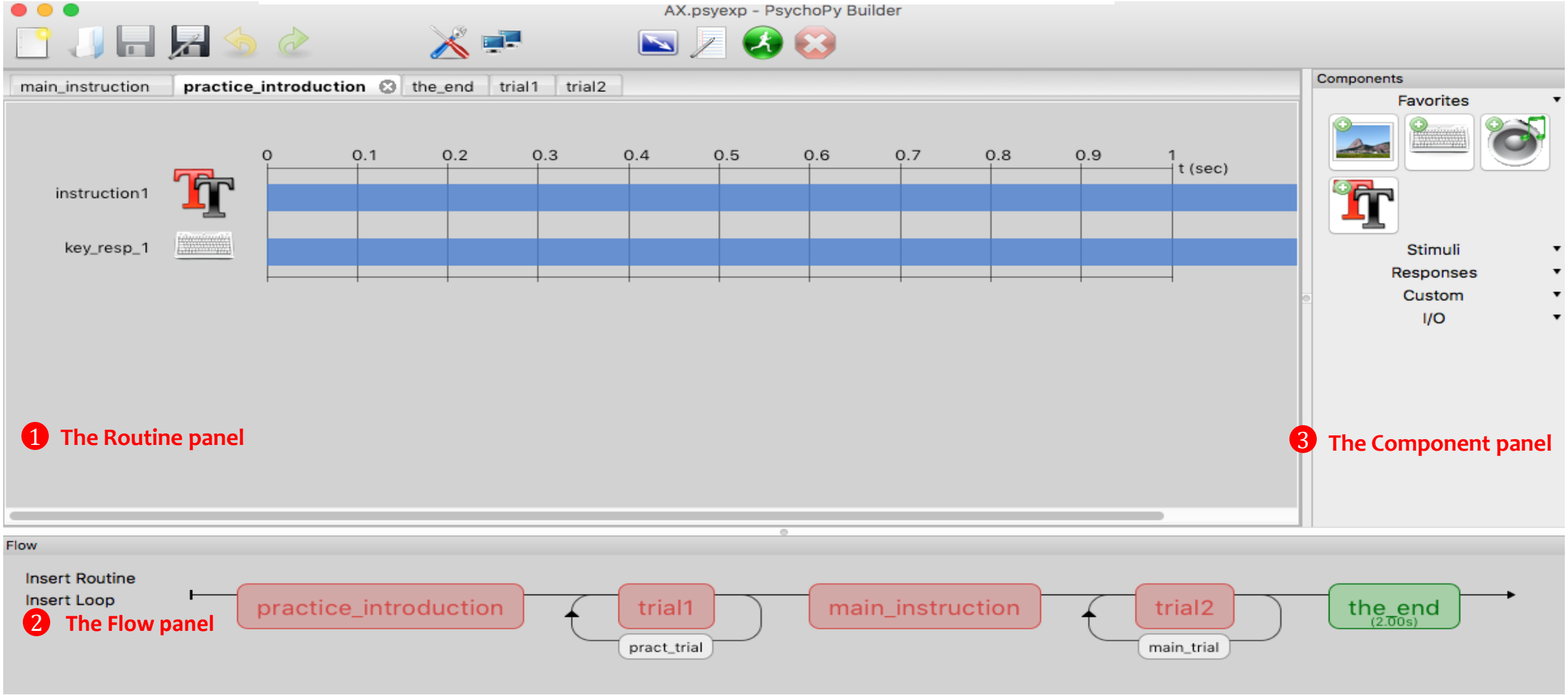
Step 3-5: Create *Routines* and *loops*



Step 6 : Run an experiment

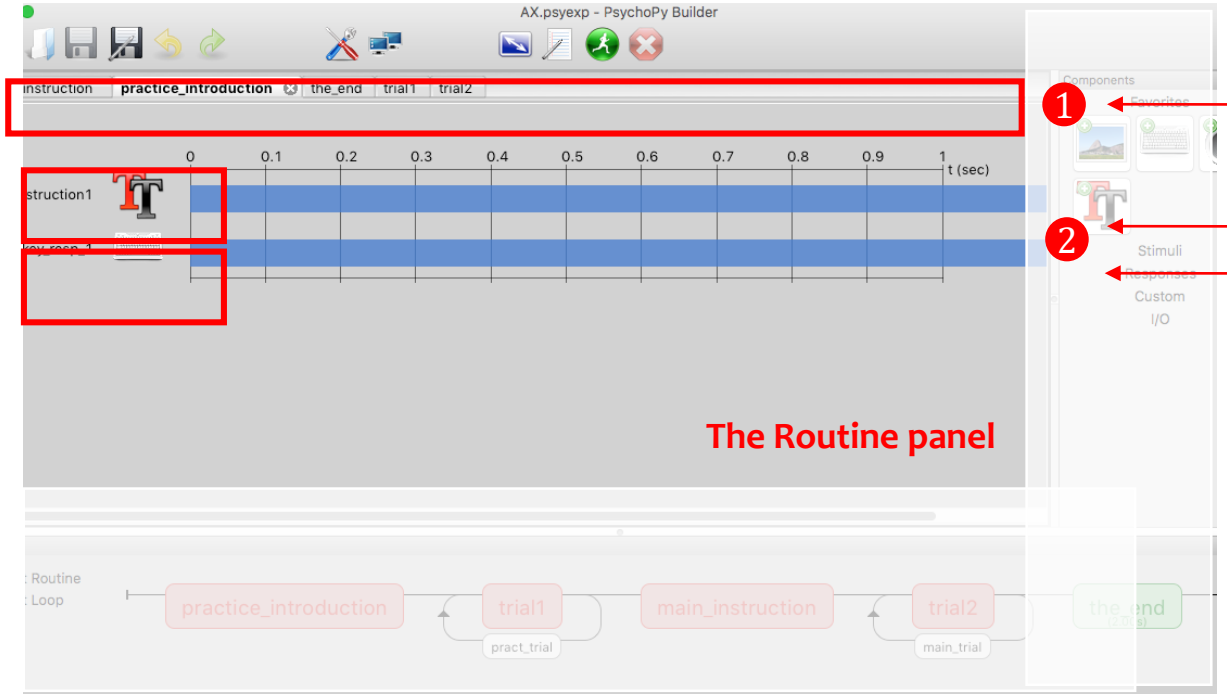


The PsychoPy *Builder* interface



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

PscychoPy the *Routine Panel*



1 These tabs show *Routines*.

2 *Text* component

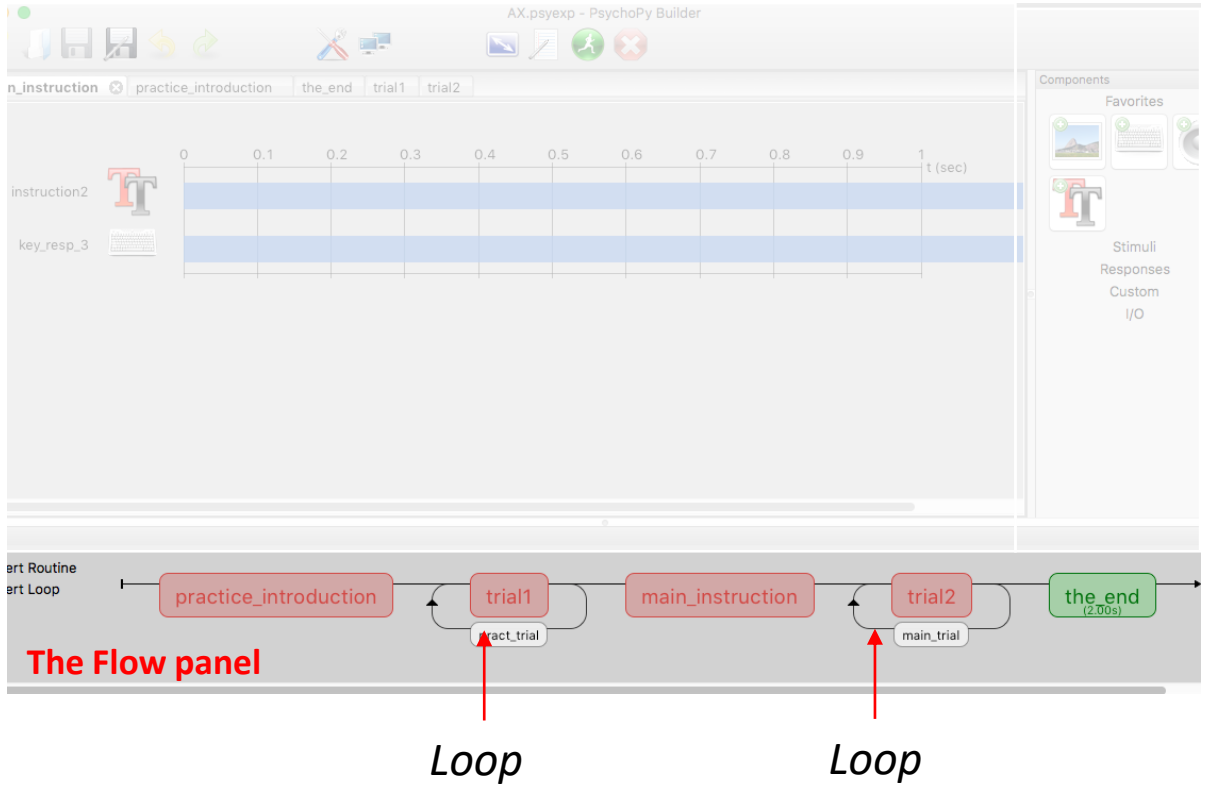
Keyboard component

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

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

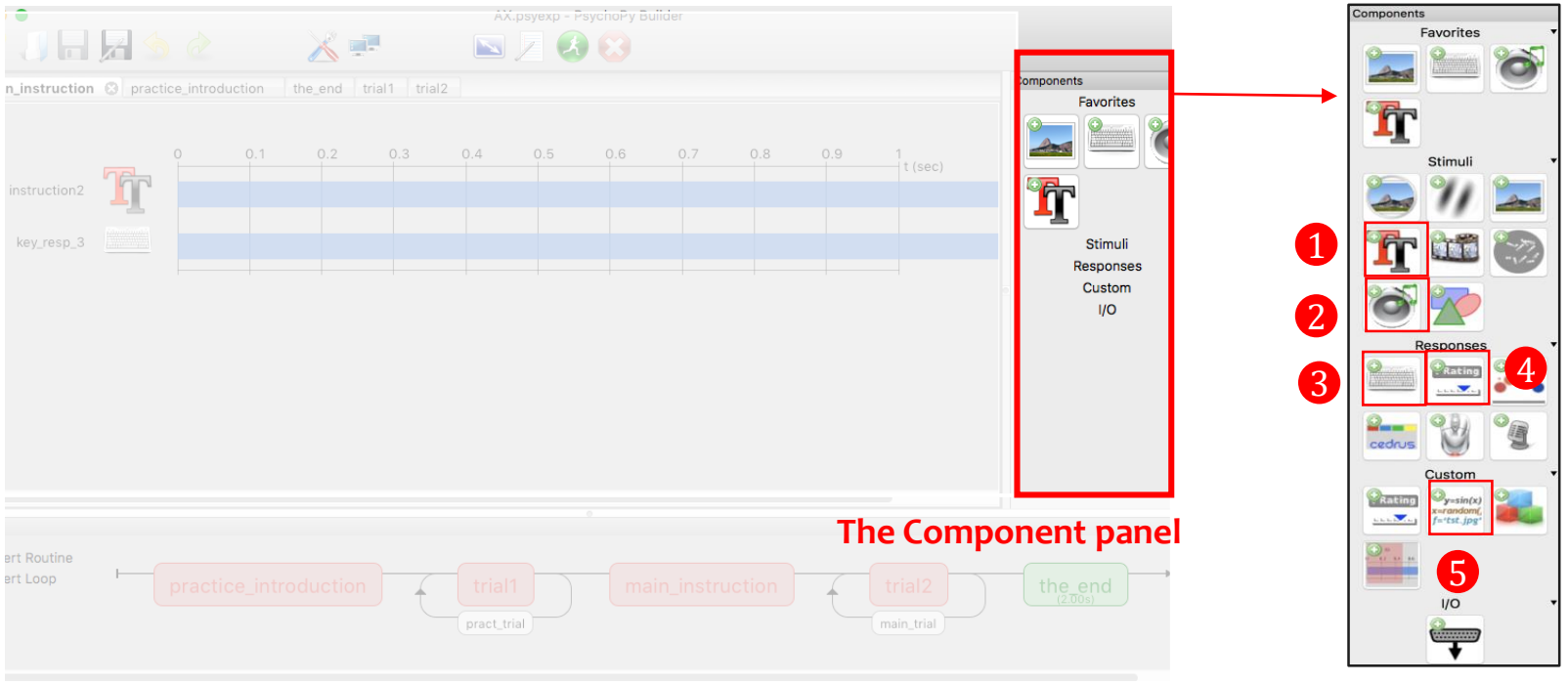
2. The Routine shown here has **a *Text*** and ***Keyboard* component.**

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.

PsychoPy *Components*



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)

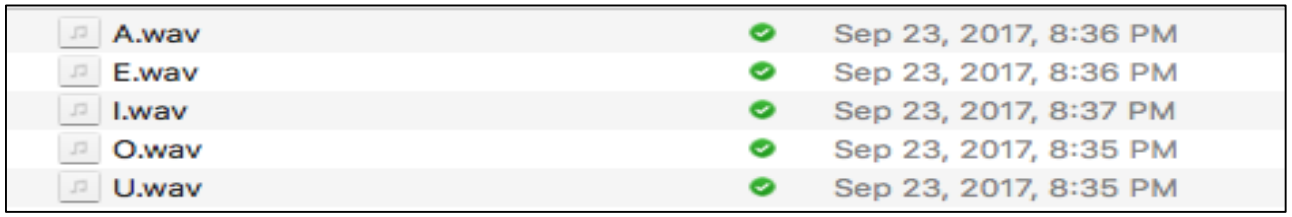
Names for the AX discrimination task

- **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		pract_trial	
***Text* component**	instruction1	samediff_msg1	instruction2	word2	text
		trial_number1		trial_number2	
***Sound* component1**		sound1_1		sound2_1	
***Sound* component2**		sound1_2		sound2_2	
***Keyboard* component**	key_resp_1	key_resp_2	key_resp_3	key_resp_4	

Step1: Create Excel files with your auditory stimuli.

Sound files are in the fold "sound"



main_stimuli. xlsx

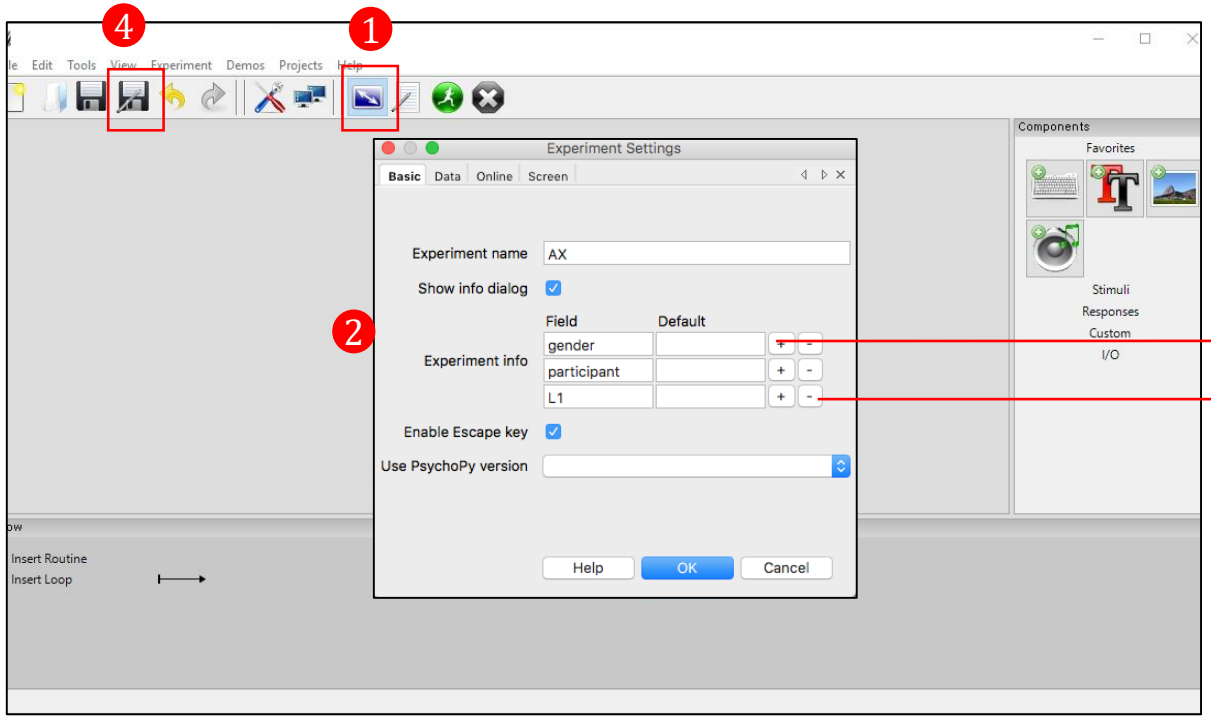
sound1	sound2	corrAns
sound/O.wav	sound/A.wav	0
sound/I.wav	sound/I.wav	1

practice_stimuli. xlsx

sound1	sound2	corrAns
sound/A.wav	sound/A.wav	1
sound/I.wav	sound/A.wav	0

- Create a folder named "AX Disciminiation" on your computer to store experiment files.
- In the ranking 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.

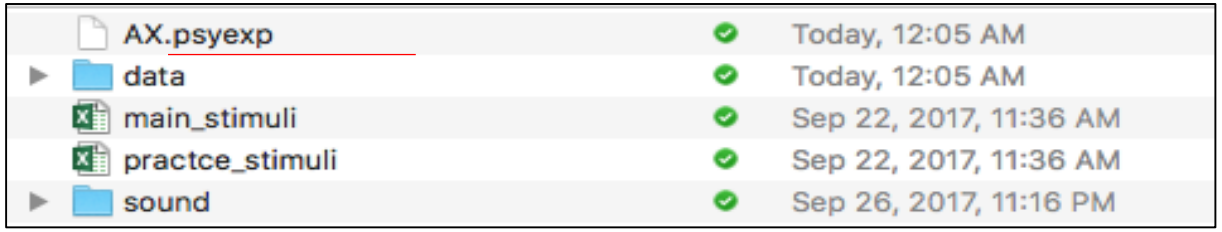
Step2: Alter experiment settings



+ To add experiment info

- To delete experiment info

4



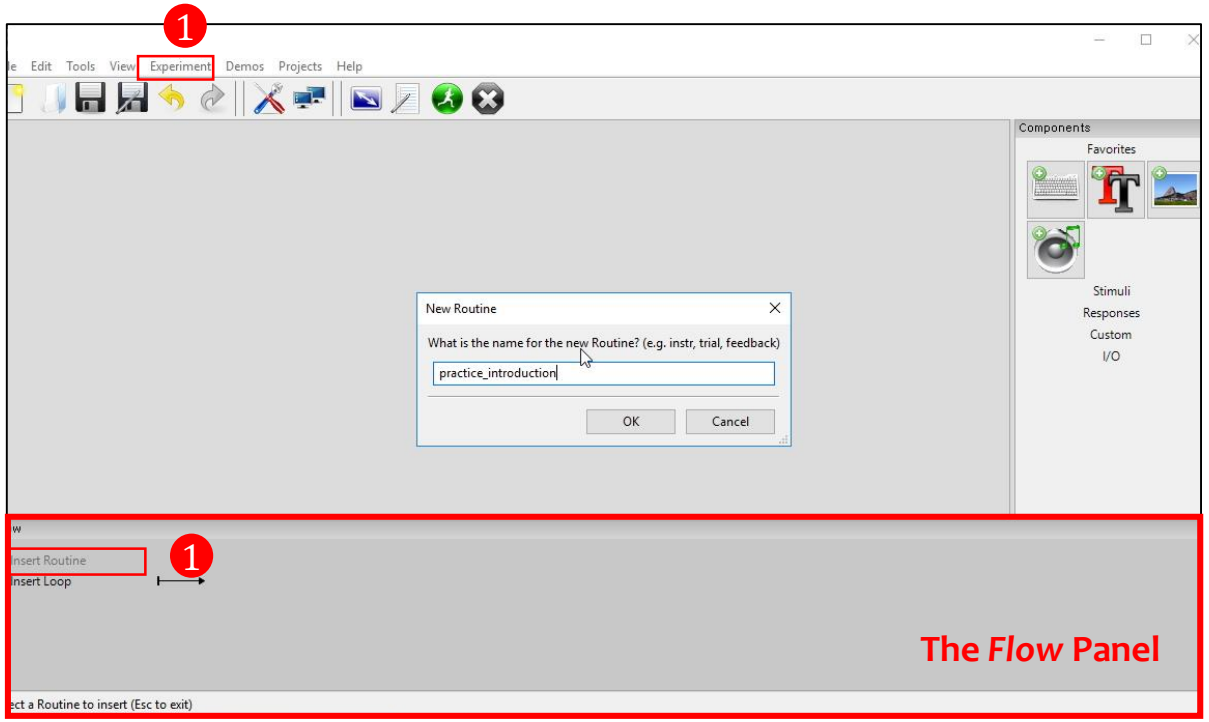
1 Click on  on the top of the screen for **experiment settings**.

2 Write an experiment name, e.g. "Rating Task", 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.

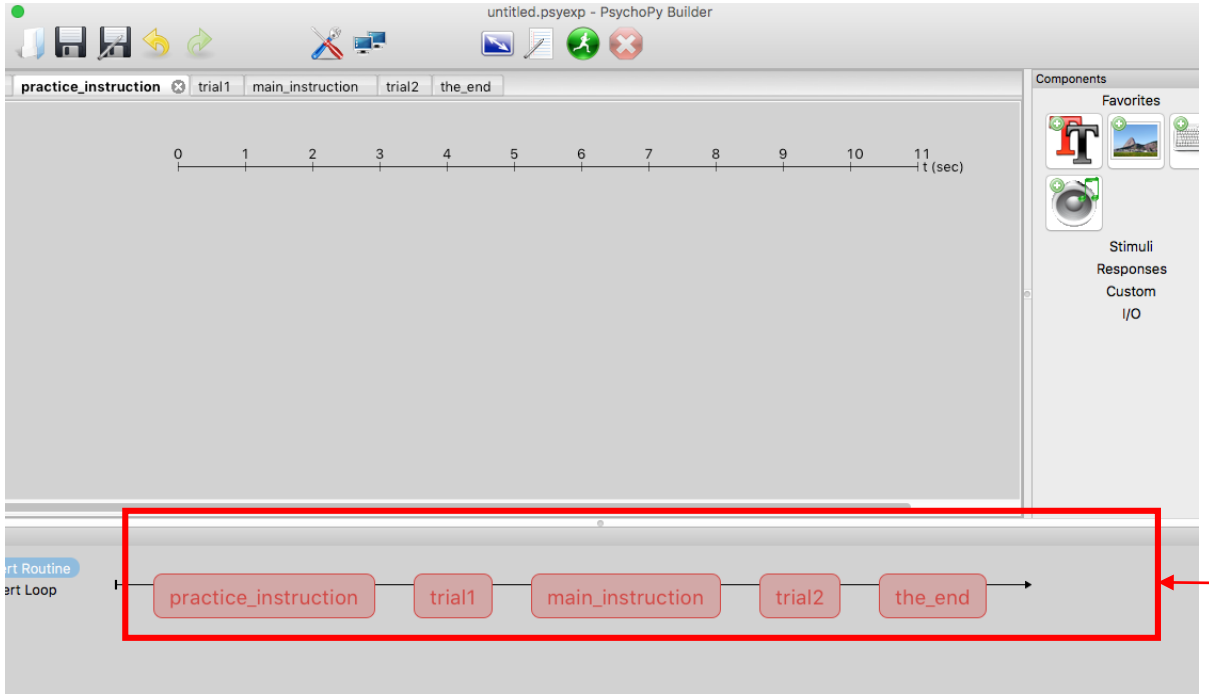
4 Save the PsychoPy file named "**AX Discrimination.psyexp**" in the same folder after setting up the rating task.

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*.

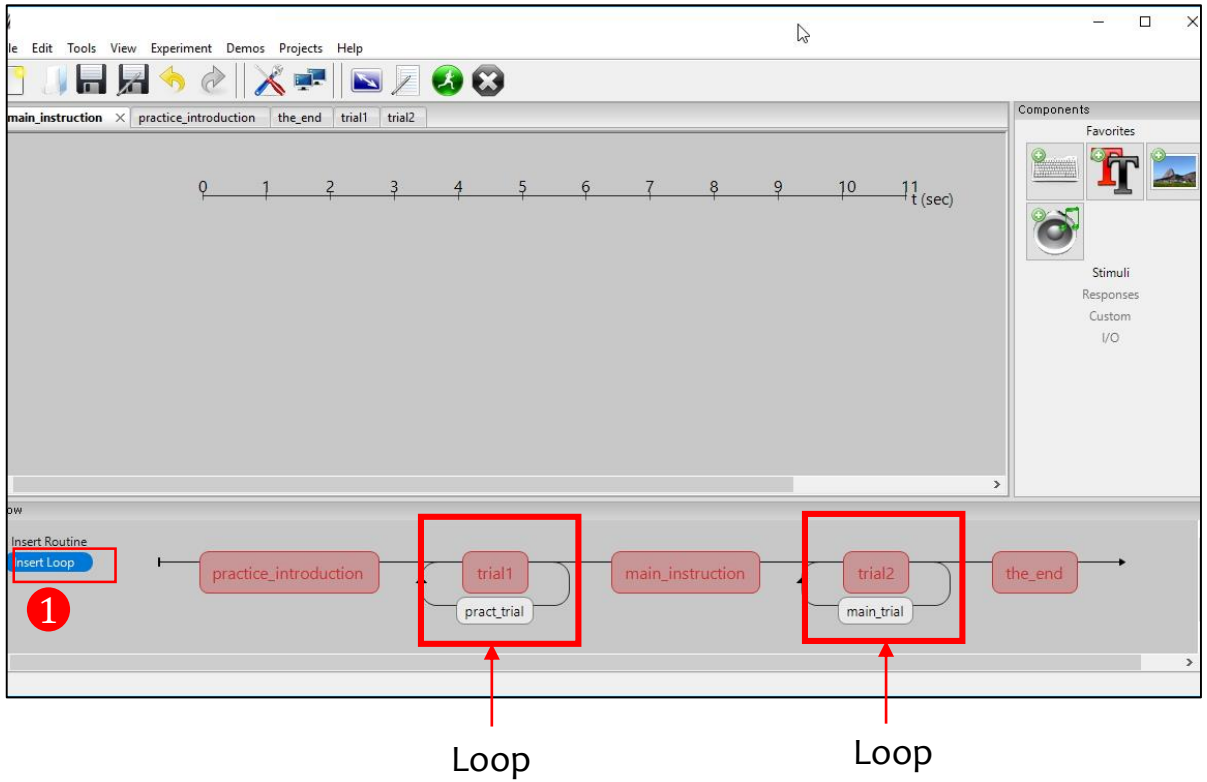
Step4: Add the *Routines* to the *Flow*

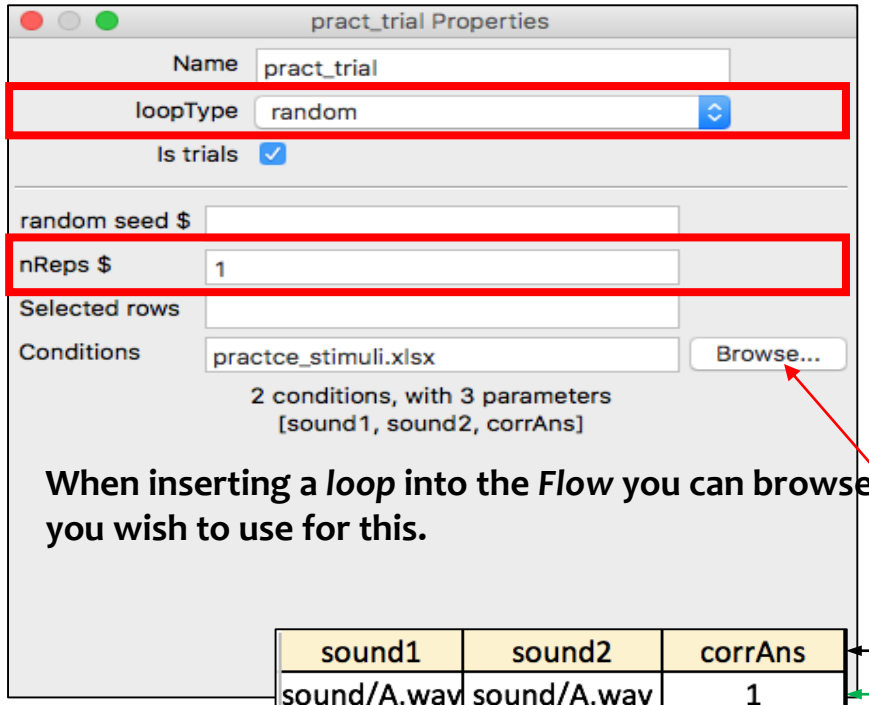


Insert five *Routines* in the *Flow* panel

- 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*





No of the repetition

3 parameters

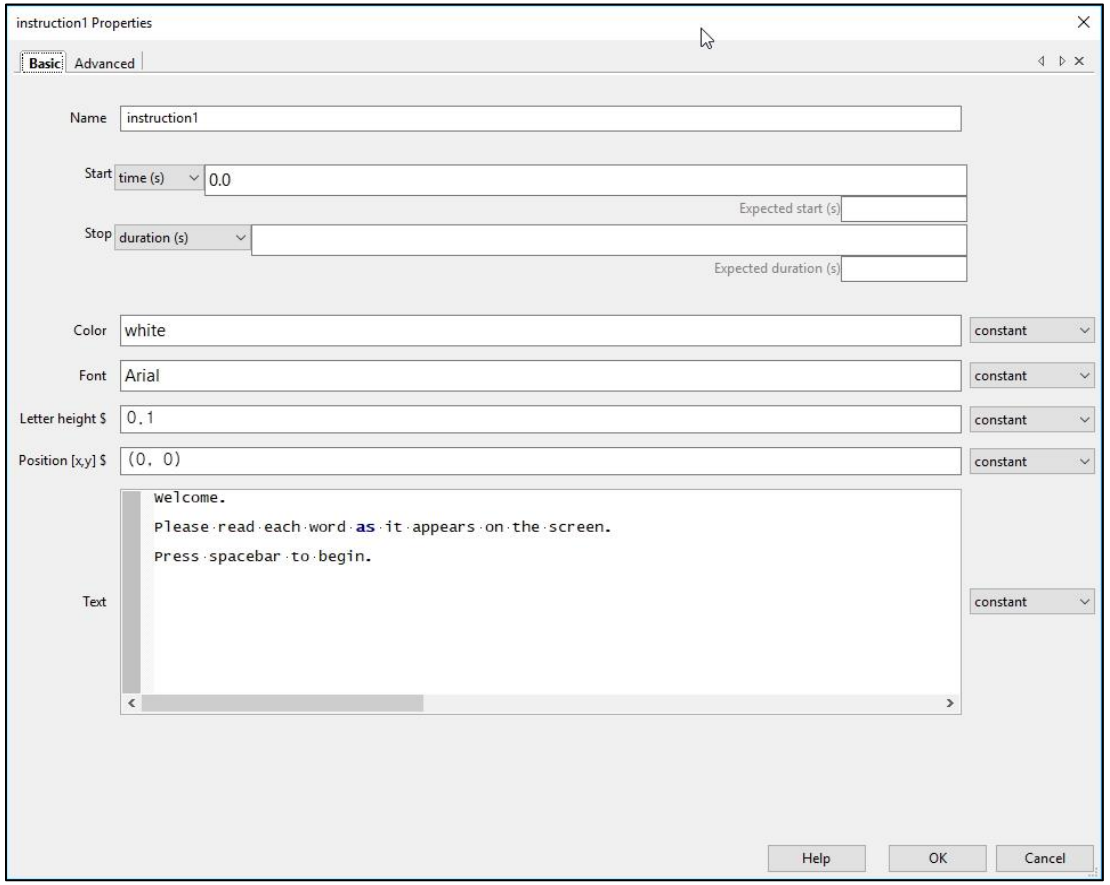
2 conditions

***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)

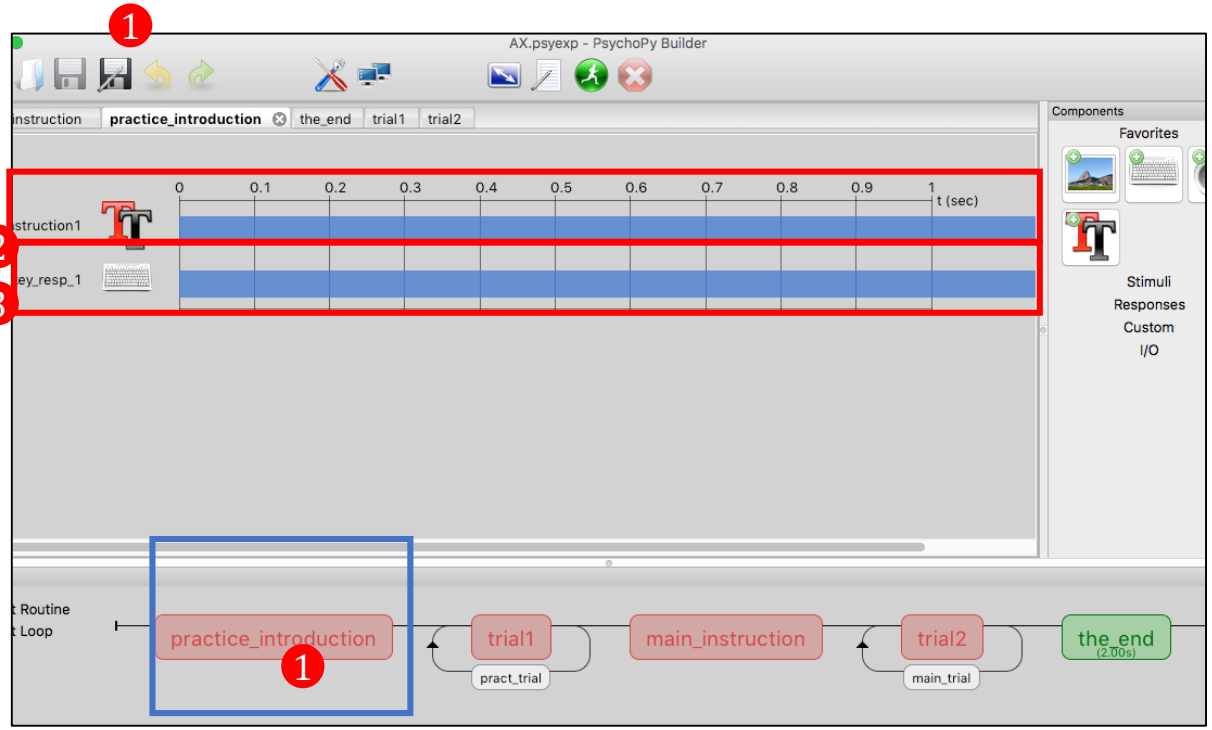
Loops and *Routines* can also be edited or removed from the *Flow* panel by clicking or right-clicking.

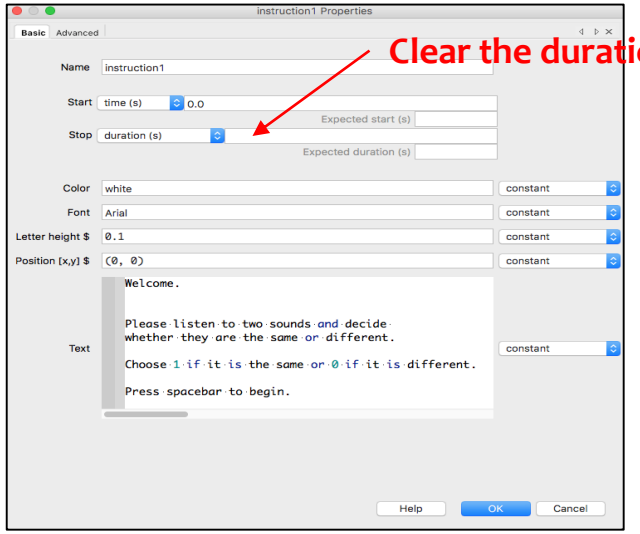
Common settings used by *Components*



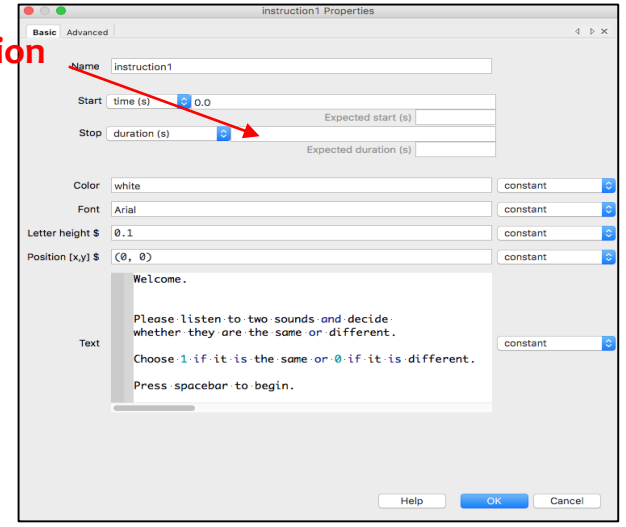
1. **Start time (seconds)** – the time after the start of the *Routine* that the stimulus will first appear.
2. **Duration (seconds)** – the length of time it will last. **If blank then the *Component* will go on indefinitely.**
3. **Font:** default font is "Arial". If you have font issues, please try 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.**

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





2 ***Text properties***

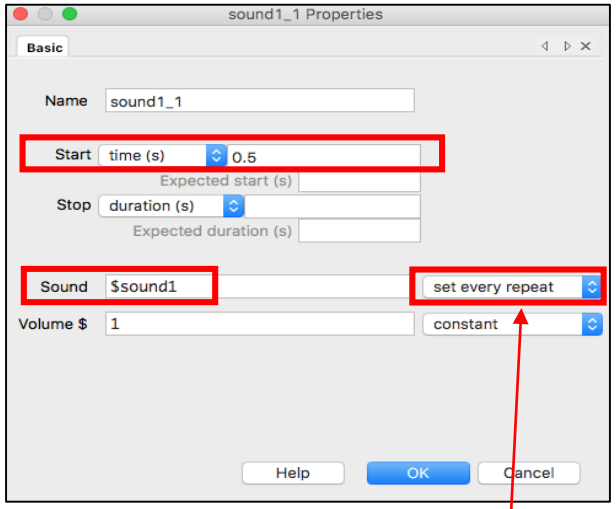


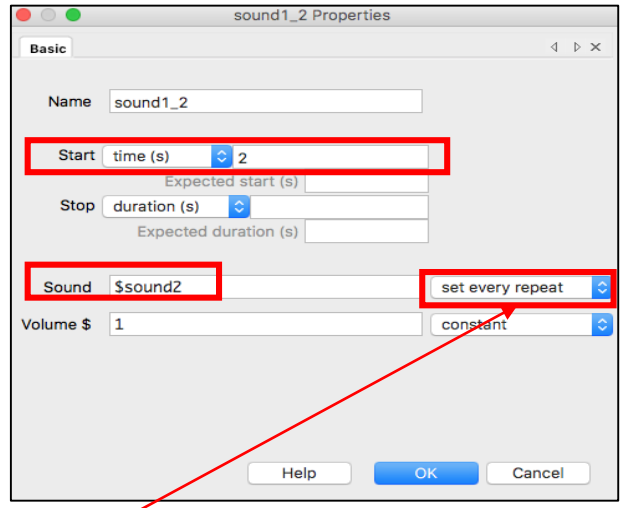
3 ***Keyboard properties***

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 ***Sound properties (sound1_1)***

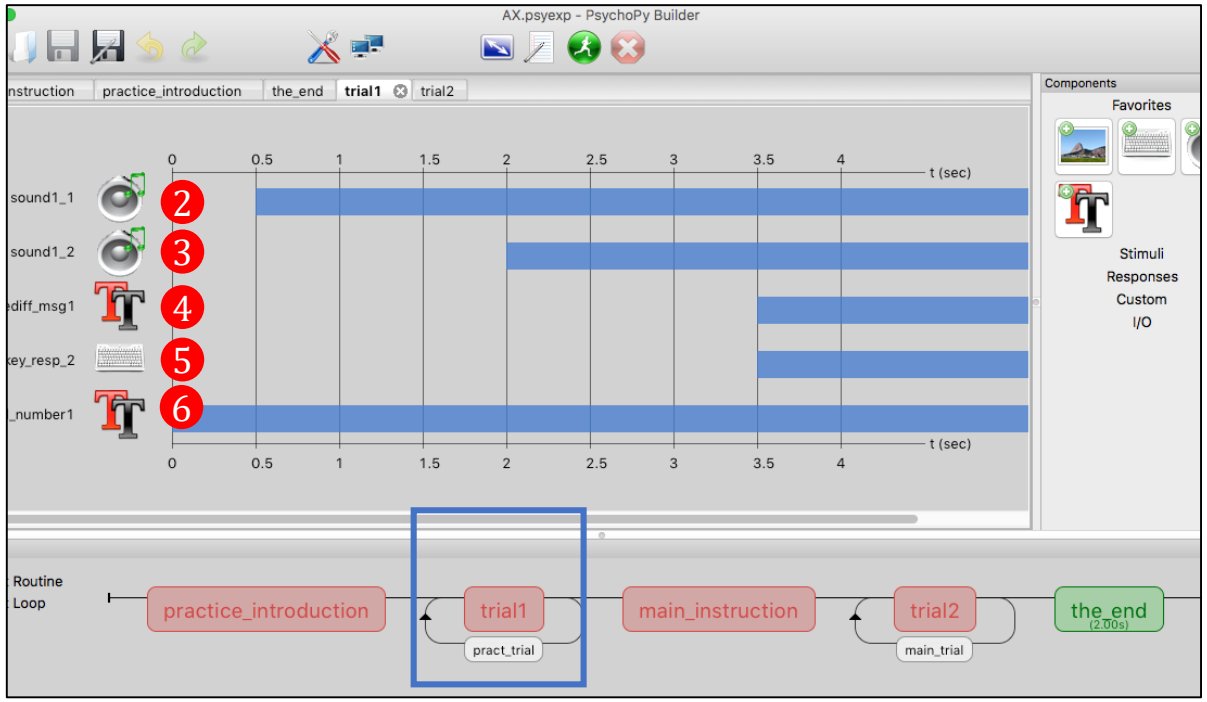
3 ***Sound properties (sound1_2)***

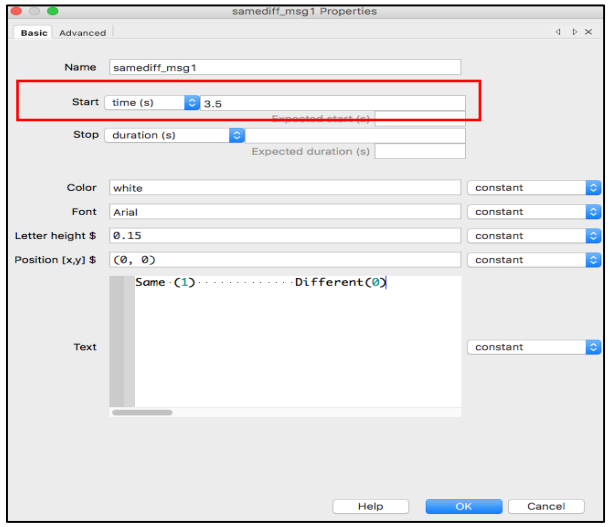
'set every repeat'
It will be updated every repeat of the *Routine*.

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

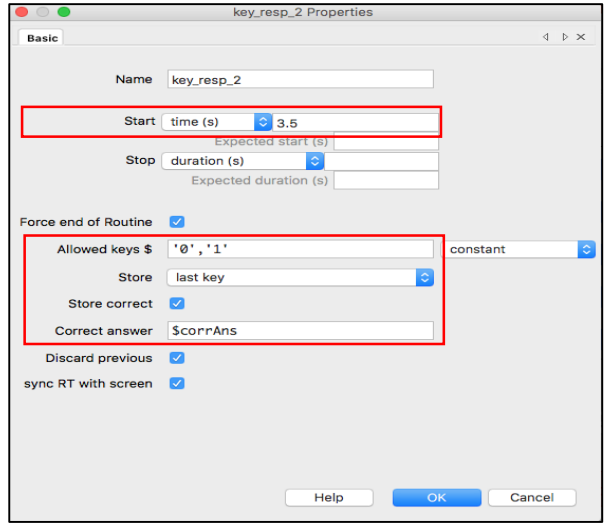
1. Select [trial1] in the tab, or in the *Flow* panel.
2. Click on in the *Component* panel and fill out the sound properties (Start time:0.5, Sound:$sound1, set every repeat)
3. Click on in the Component panel and fill out the sound properties (Start time:2, Sound:$sound2, set every repeat)

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





4 ***Text properties (samediff_msg1)***



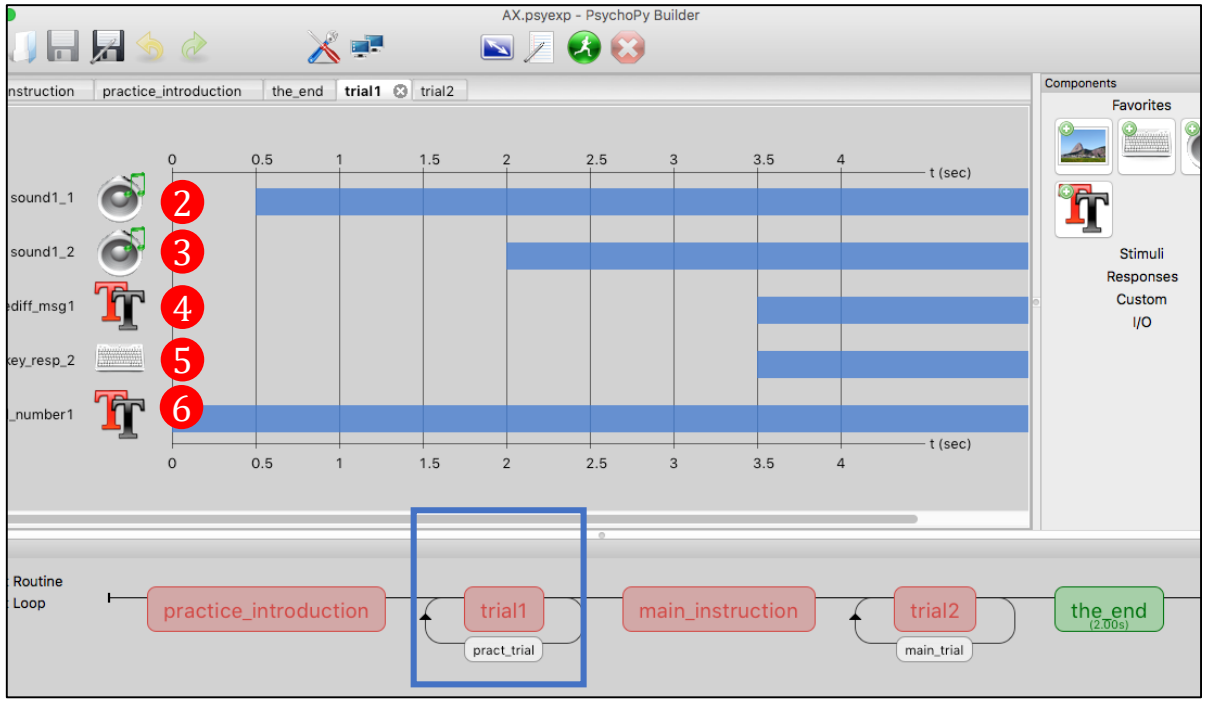
5 ***Keyboard properties (key_resp_2)***

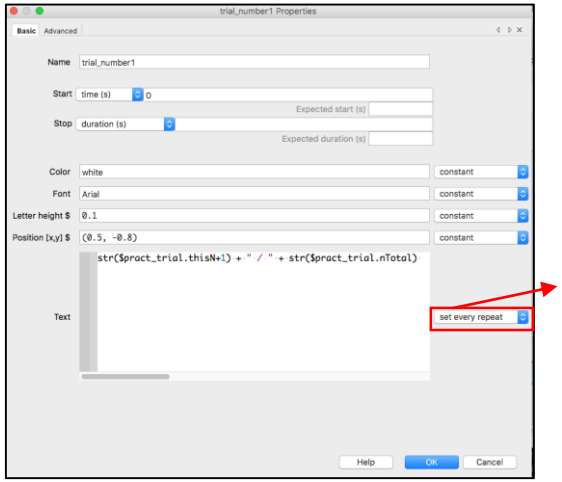
For the *Routine* [trial1] , two *Sound* components, two *Text* component and a *Keyboard* component are needed.

4 Click on  in the *Component* panel and fill out the text properties: Start time: 3.5, Font: Arial, Letter Height: 0.15, Position: (0,0)

5 Click on  in the *Component* panel and fill out the keyboard properties: Start time:3.5, Allowed keys: ‘0’, ‘1’ , Store: last key, Correct answer: $corrAns

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





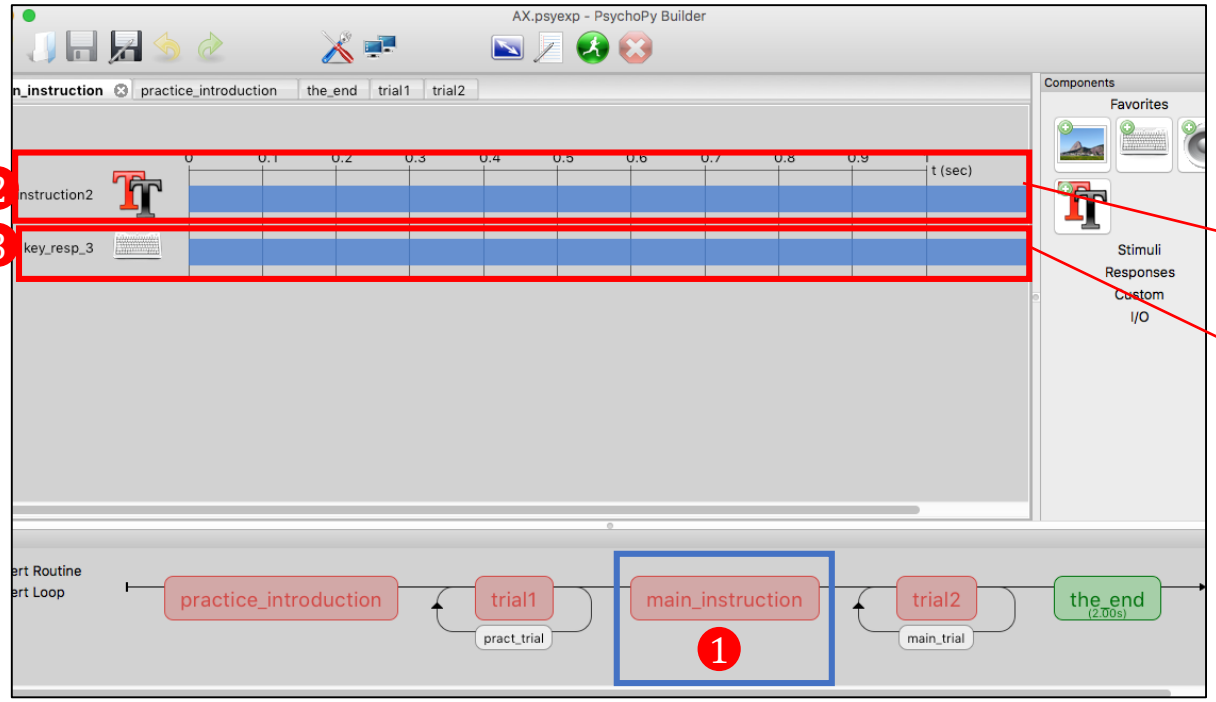
'set every repeat'
It will be updated every repeat of the *Routine*.

❻ ***Text properties (Trial_number1)***

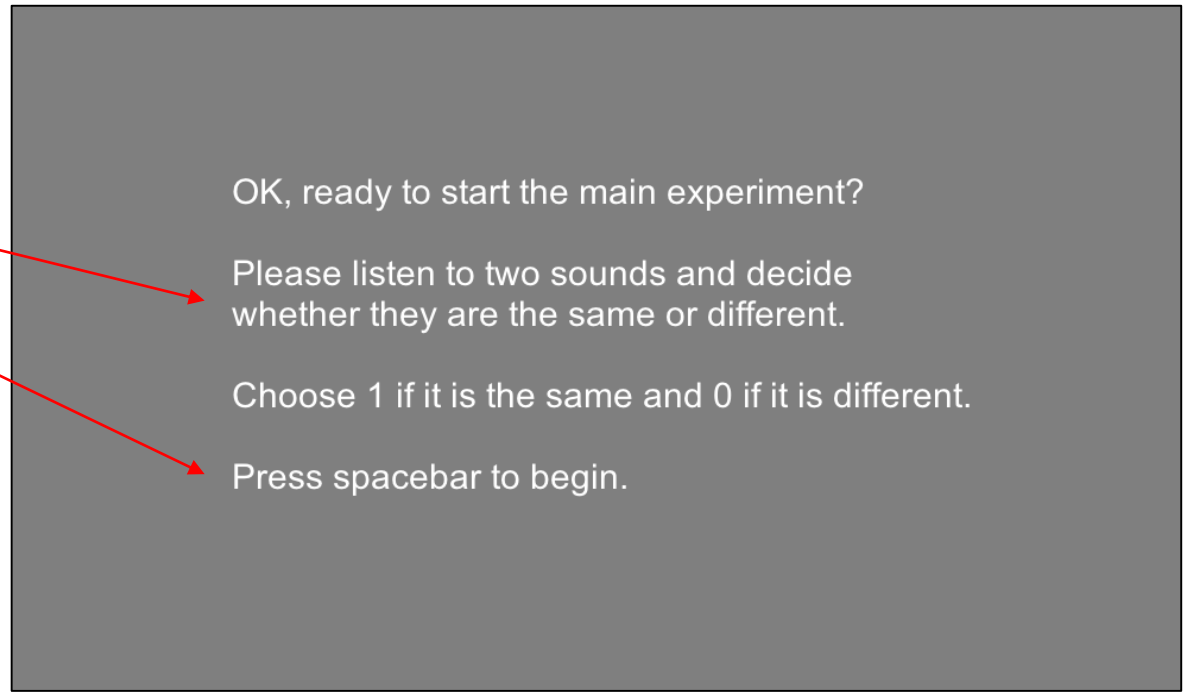
For the *Routine* [trial1] , two *Sound* components, two *Text* component and a *Keyboard* component are needed.

❻ Click on T in the *Component* panel and fill out the text properties: Font: Arial, Letter Height: 0.1, Position: (0.5, -0.8), str($pract_trial.thisN+1) + " / " + str($pract_trial.nTotal)

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



Screenshot in PsycoPy Builder view

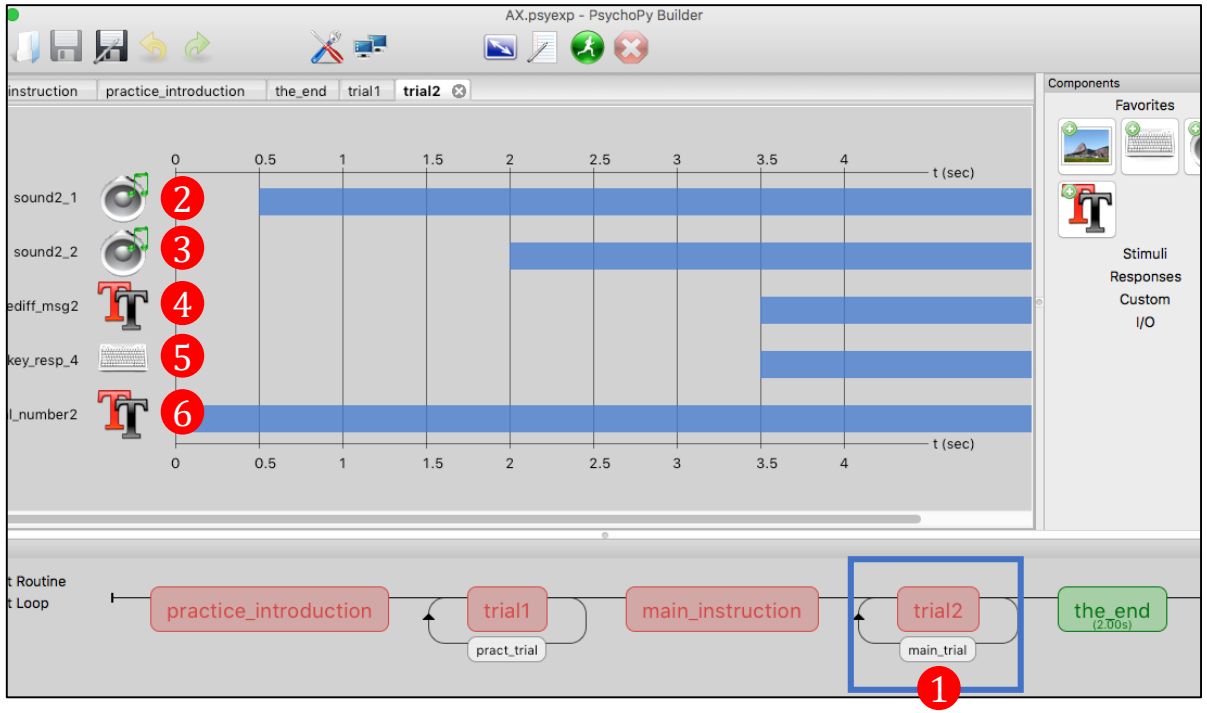


Screenshot in the experiment

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 PsycoPy Builder view

Screenshot in the experiment

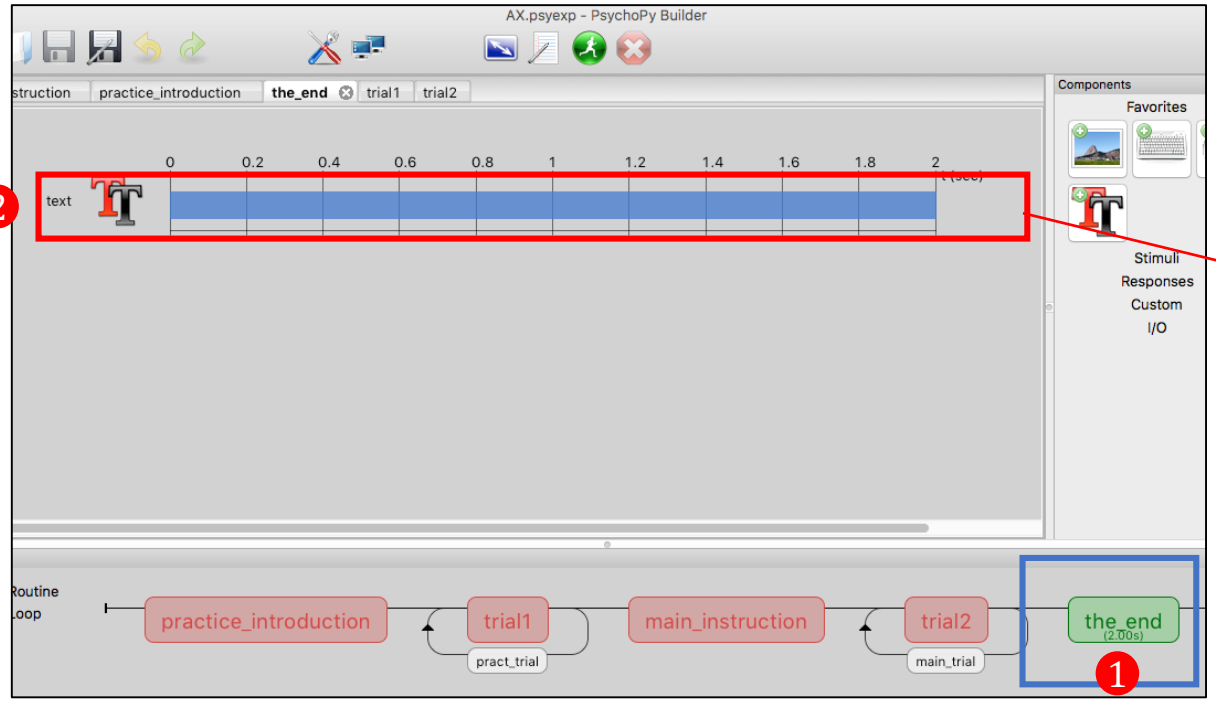
Setting up the *Routine* [trial2] is the same as the *Routine* [trial1].

❶ Select [trial2] in the tab, or in the *Flow* panel.

❷❸❺ Add two *Text* components (samediff_msg2, trial_number2) in the *Routine* panel and fill out the *Text* properties for each.

❹❻ Add two *Sound* component (sound2_1, sound2_2) and a keyword 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]



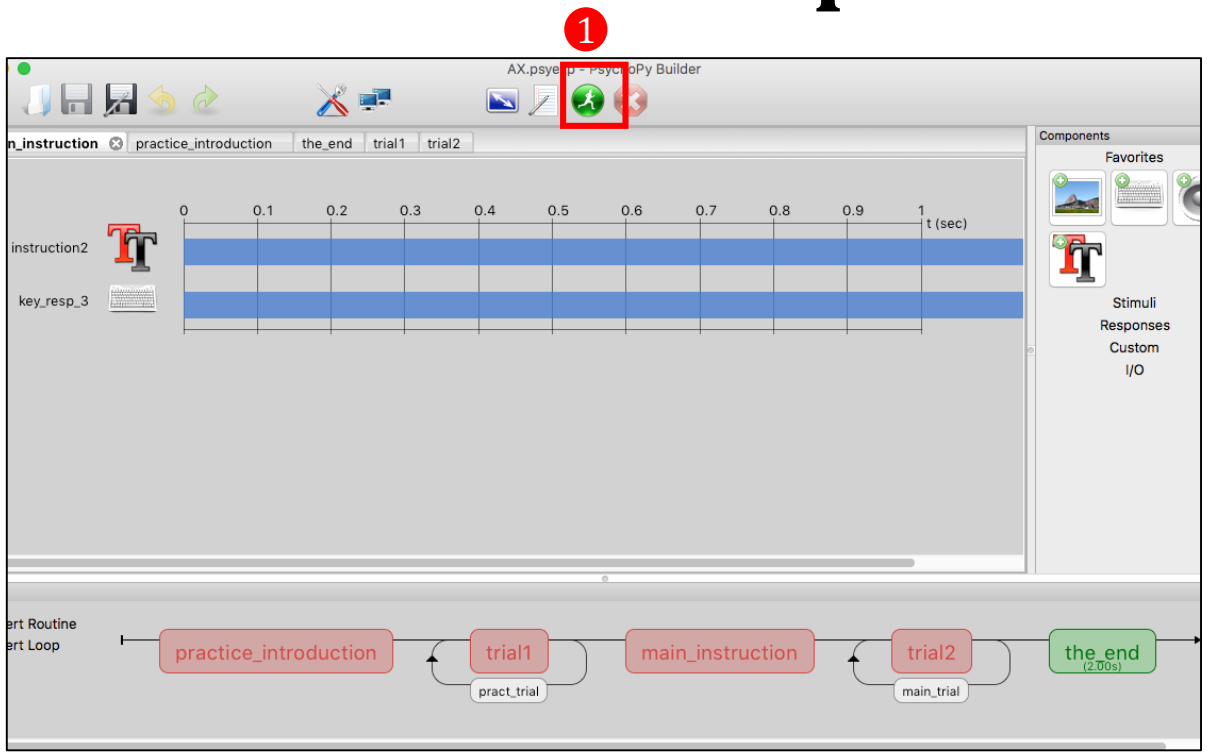
Screenshot in PsycoPy Builder view

Screenshot in the experiment

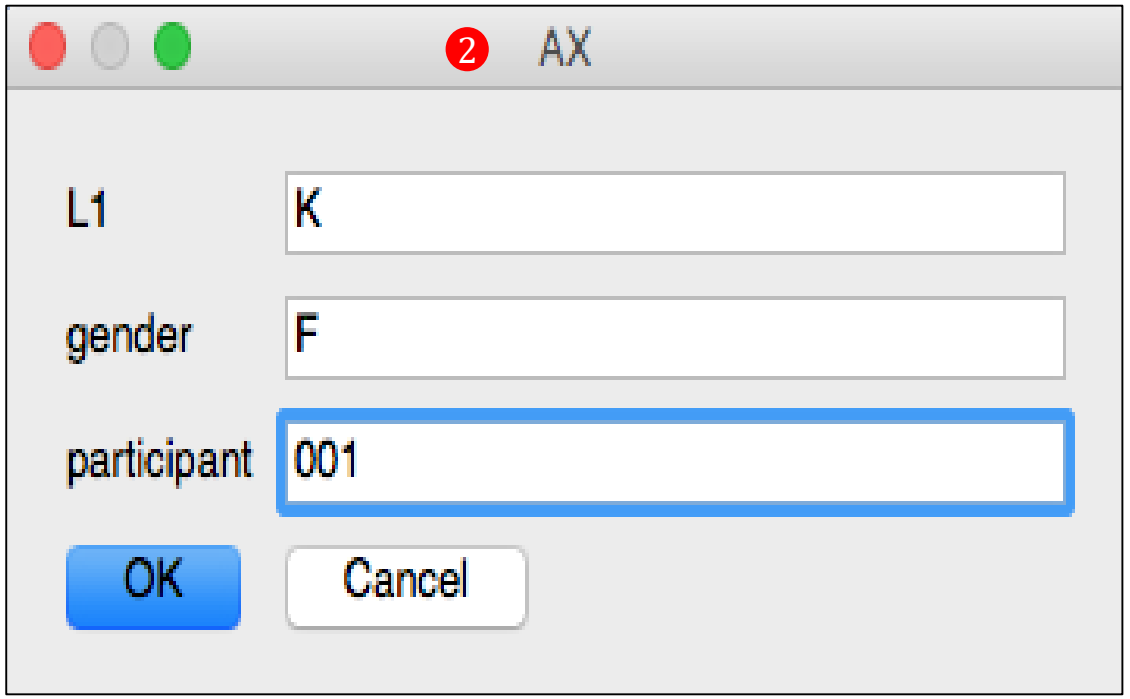
1. Select the *Routine* [the_end] in the tab, or in the *Flow* panel.

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 PsycoPy 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 the AX discrimination task in which a participant (Gender: F, L1: Korean) judge whether two sounds are the same or difference.

sound1	sound2	corrAns	_trial.this	_trial.this	ct_trial.th	_trial.this	_trial.this	_trial.this	in_trial.th	_trial.this	_resp_2.k	y_resp_2.c	y_resp_2	_resp_4.k	y_resp_4.c	y_resp_4	participan	gender	frameRate	expName	L1	date
sound/A.wav	sound/A.wav	1	0	0	0	0					1	1	9.0093				1	F	60.027	AX	K	2017_Oct_19_0003
sound/l.wav	sound/A.wav	0	0	1	1	1					1	0	0.8871				1	F	60.027	AX	K	2017_Oct_19_0003
sound/l.wav	sound/l.wav	1					0	0	0	1				1	1	25.089	1	F	60.027	AX	K	2017_Oct_19_0003
sound/O.wav	sound/A.wav	0					0	1	1	0				1	0	1.1536	1	F	60.027	AX	K	2017_Oct_19_0003
sound/l.wav	sound/l.wav	1					1	0	2	1				1	1	0.9187	1	F	60.027	AX	K	2017_Oct_19_0003
sound/O.wav	sound/A.wav	0					1	1	3	0				1	0	0.5367	1	F	60.027	AX	K	2017_Oct_19_0003

Useful websites & 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=VV6qhuQgsiI
  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).