

BUILDING EXPERIMENTS IN



By Johnny Lau

Modified on 28 Mar, 2019

EXPERIMENTAL SOFTWARES

- **E-prime**
- **Superlab**
- **Presentation**
- **JAVA / PHP (mainly for online experiments?)**
- **Psychtoolbox/ Cogent (Matlab-based)**
- **PsychoPy**
- **.....**

PSYCHOPY - ADVANTAGES

- Open Source (FREE)
- Run in Python (open source as well)
- Cross-Platform (Windows, Mac, Linux)
- Quite user-friendly: unique interface
 - Builder Interface
 - Coder Interface

PSYCHOPY – WEAKNESSES?

- **Graphical interface is limited:** for the more advanced experiment > you have to script it yourself
- **Timing of Audio Stimuli is NOT perfect** (sound latency issues)

Potential solution:

<https://discourse.psychopy.org/t/psychopy-2-sound-latency-an-update/4591>

PSYCHOPY - FEATURES

- **Builder (GUI)**
- **Coder**

```
from psychopy import visual, core
win = visual.Window([400,400])
message = visual.TextStim(win, text='hello')
message.setAutoDraw(True) # automatically draw every frame
win.flip()
core.wait(2.0)
message.setText('world') # change properties of existing stim
win.flip()
core.wait(2.0)
```

PSYCHOPY – BUILDER INTERFACE

The screenshot displays the Psychopy Builder interface, which is used for creating experiments. The interface is divided into three main sections, each highlighted with a red rounded rectangle and a label:

- Routines:** This section is located in the top-left area. It features a timeline labeled 't (sec)' from 0 to 1.0. Two routines are currently added: 'instr_txt' (represented by a large 'T' icon) and 'instr_resp' (represented by a document icon). The timeline is divided into 10 segments of 0.1 seconds each.
- Components:** This section is located in the top-right area. It contains a 'Favorites' list with icons for a text box, a document, a landscape image, and a camera. Below this are categories for 'Stimuli', 'Responses', and 'Custom'. The 'Custom' category is expanded, showing a 'Rating' component, a 'y=sin(x)' component, an 'x=random()' component, and an 'f=tst.jpg' component. There is also an 'I/O' section with a component icon.
- Flow:** This section is located at the bottom of the interface. It shows a flow diagram with two components: 'Instructions' (a red box) and 'trial (3.50s)' (a green box). Arrows indicate the flow from 'Instructions' to 'trial'. On the left side of the flow diagram, there are buttons for 'Insert Routine' and 'Insert Loop'.

PSYCHOPY – EXERCISE

STROOP TASK

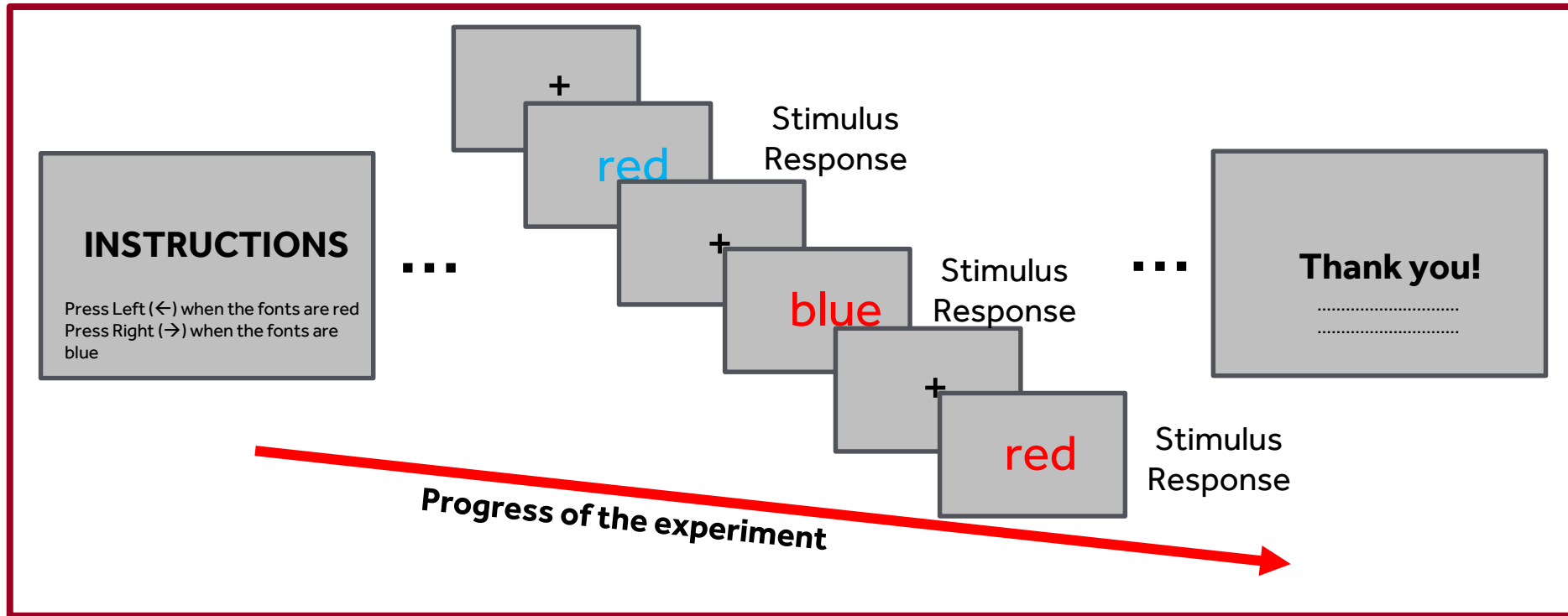
Task1 (~25 mins)

- **Familiarising with the software**
- **Basic control**

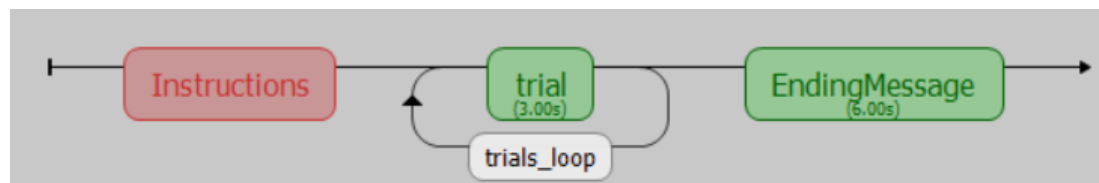
- Task 2 (~35mins)
- Exploring other features
- Coding

TASK 1

ILLUSTRATION OF THE EXPERIMENTAL TASK

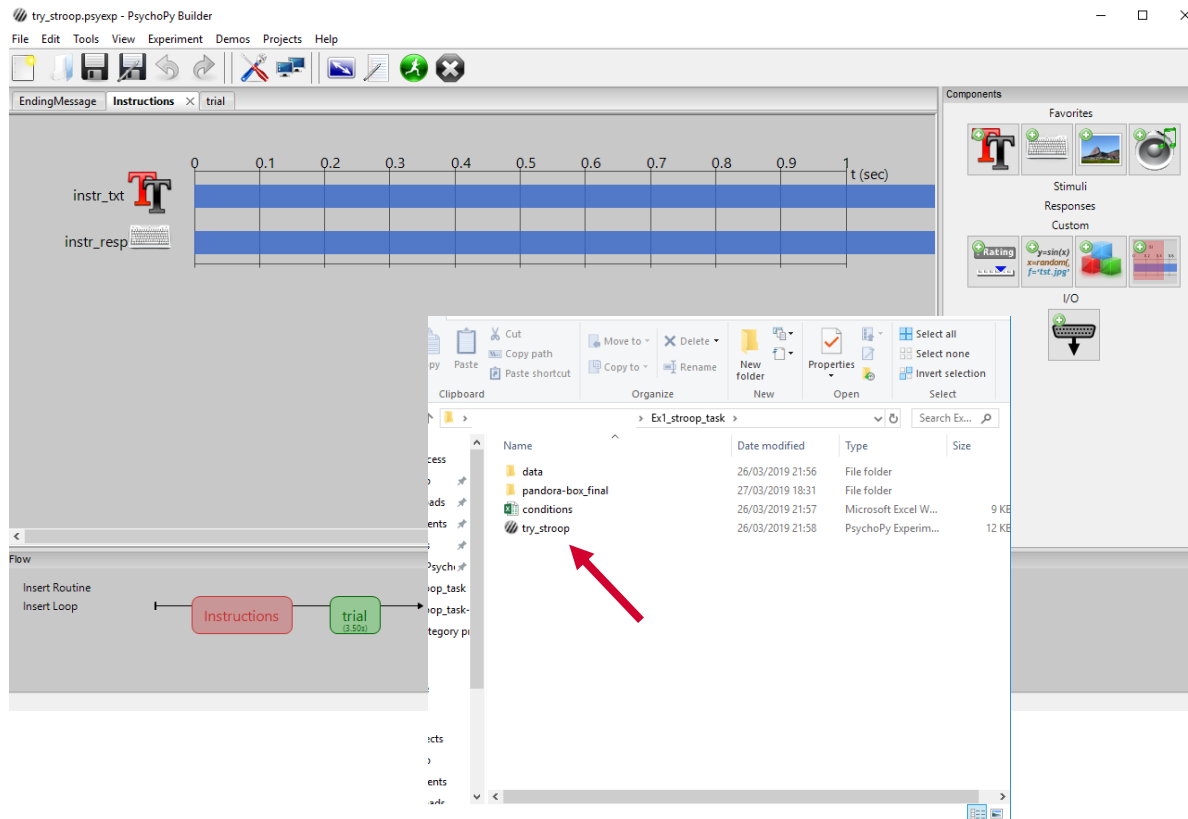


How it would be represented in PsychoPy:



PREPARATION

- In folder 'Ex1_stroop_task', open the experiment file 'try_stroop.psyexp'
- If you want to, you can drag this whole folder onto Desktop (or to any directory where you want to keep this task).
- Remember to save your work regularly from now on.



INSTRUCTIONS ROUTINE

(This has already been created for you)

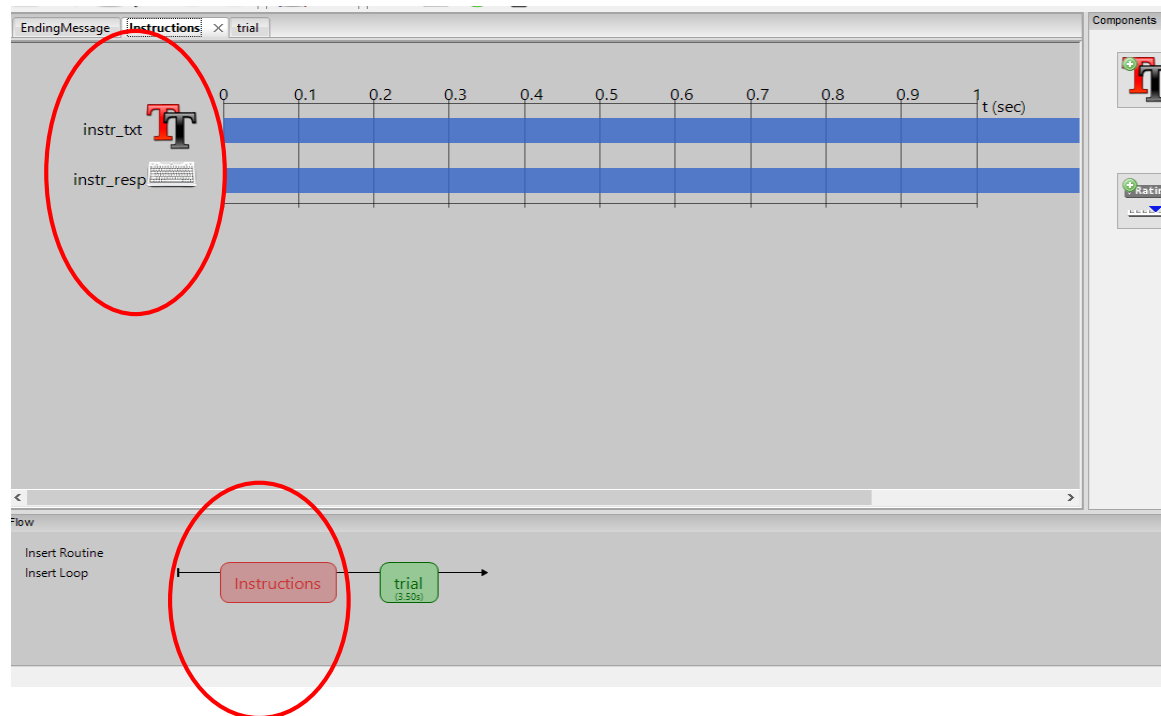
Thanks for taking part. In this task, you need to focus and report whether the COLOUR of the fonts matches the WORD itself.

Press Left (<) when the COLOUR of the fonts matches the WORD

Press Right (->) when the COLOUR of the fonts DOESN'T match the WORD

Now press SPACE bar to continue

In the 'Instructions' routine, there are two components: Text & Keyboard

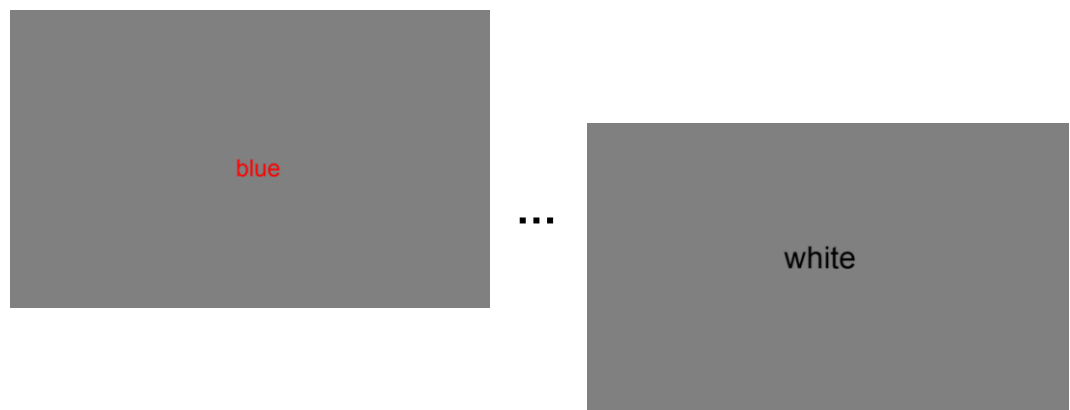


Text: for presenting instruction texts

Keyboard: create a response key (space bar) for participants to proceed

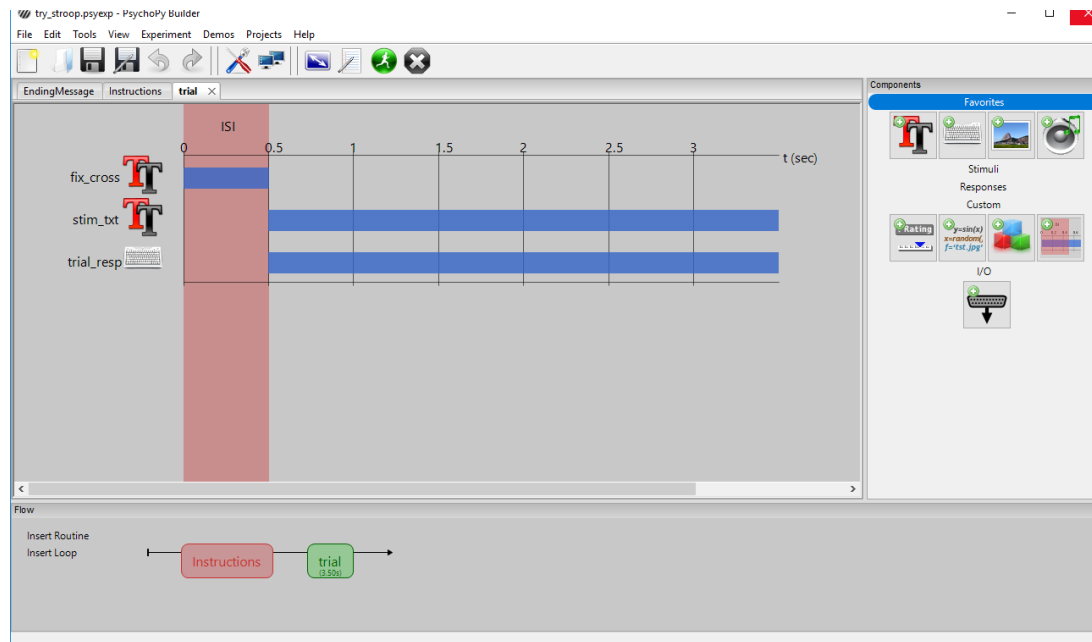
Take some time to examine them...

THE MAIN TRIALS



THE MAIN TRIALS

- This is the core part of the experiment
- In each trial, participants see 1) a 'fixation', 2) a stimulus item, and then 3) make a response



Before looking into each component in depth, let's move onto the next part (we will come back to this later)

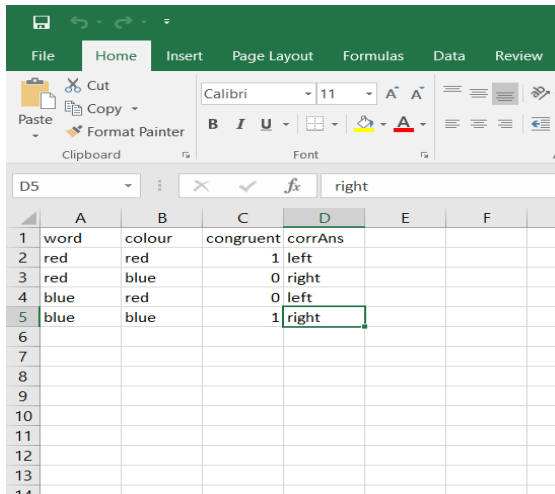
CREATING THE CONDITION FILE

- Right now your PsychoPy experiment has no information about what to use when filling the variables (i.e. what stimulus to present from trial to trial).

	Examples	
Congruent (1)	red	blue
Incongruent (0)	red	blue

Modifying the condition file on your own!

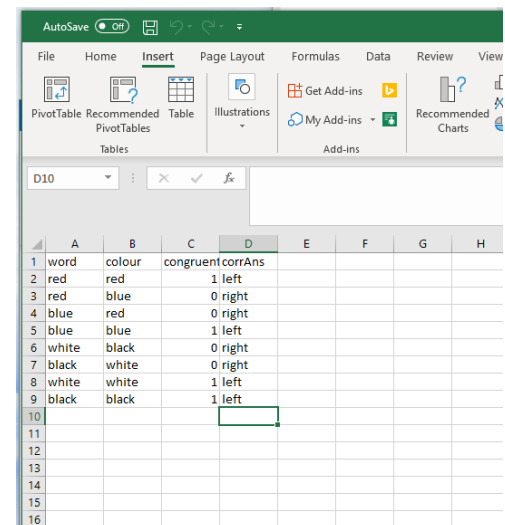
- Here we'll give PsychoPy that information
- Now, open the excel file (.xls/.csv) named 'conditions' inside the 'Ex1_stroop_task' folder. This excel file is better to be stored in the same directory as your .psyexp file.
- In the 'conditions' file, there are four columns: **word**, **colour**, **congruent**, **corrAns** (Note: no space in the names).



	A	B	C	D
1	word	colour	congruent	corrAns
2	red	red	1	left
3	red	blue	0	right
4	blue	red	0	left
5	blue	blue	1	right

- The 'word' column: the written words to be shown on the screen
- The 'colour' column specifies the colour of the fonts
- The 'congruent' column: '1' represents when 'word' and 'colour' are congruent (match); '0' when they are not
- The corrAns: the correct responses that should be made to the stimulus (*Remember what we said in the instructions? Left for MATCH; Right for MISMATCH)

Now, I want you to modify this 'conditions' file by adding more exemplars (see the example on the right) and then save the file

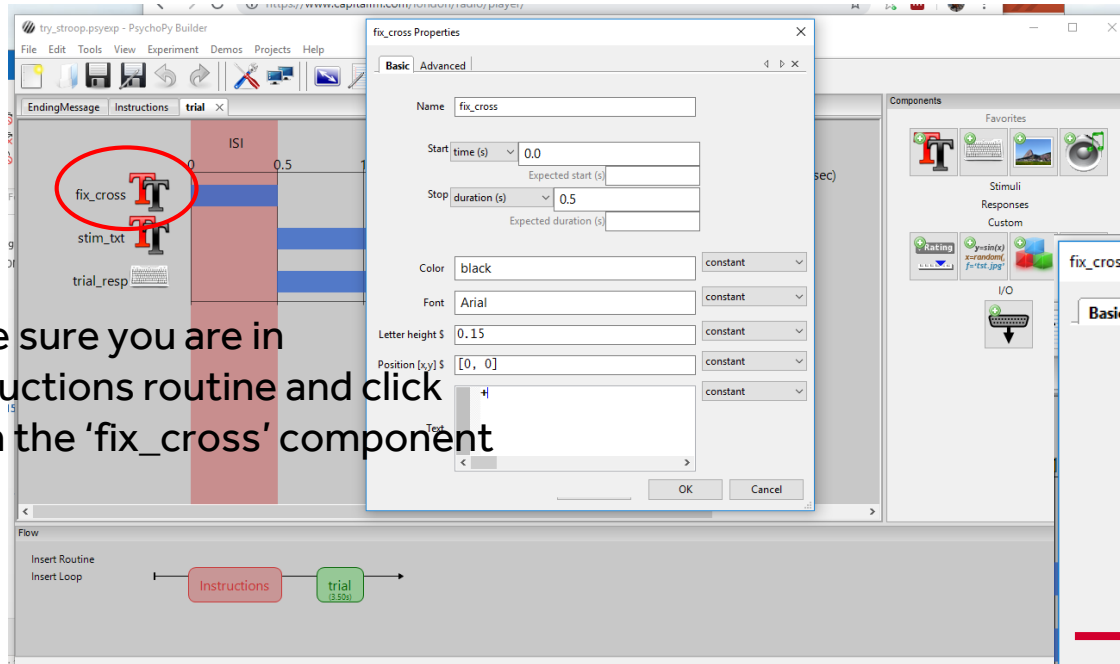


	A	B	C	D
1	word	colour	congruent	corrAns
2	red	red	1	left
3	red	blue	0	right
4	blue	red	0	right
5	blue	blue	1	left
6	white	black	0	right
7	black	white	0	right
8	white	white	1	left
9	black	black	1	left

PREPARING FOR STIMULUS PRESENTATION

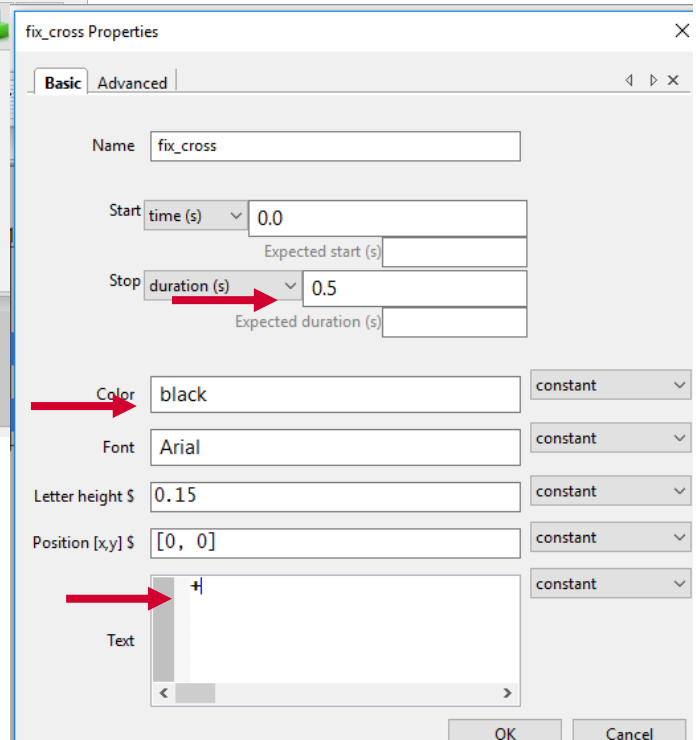
Now, return to PsychoPy and prepare a 'template' for stimulus presentation. We want to start a trial with showing a fixation cross for 0.5sec

1

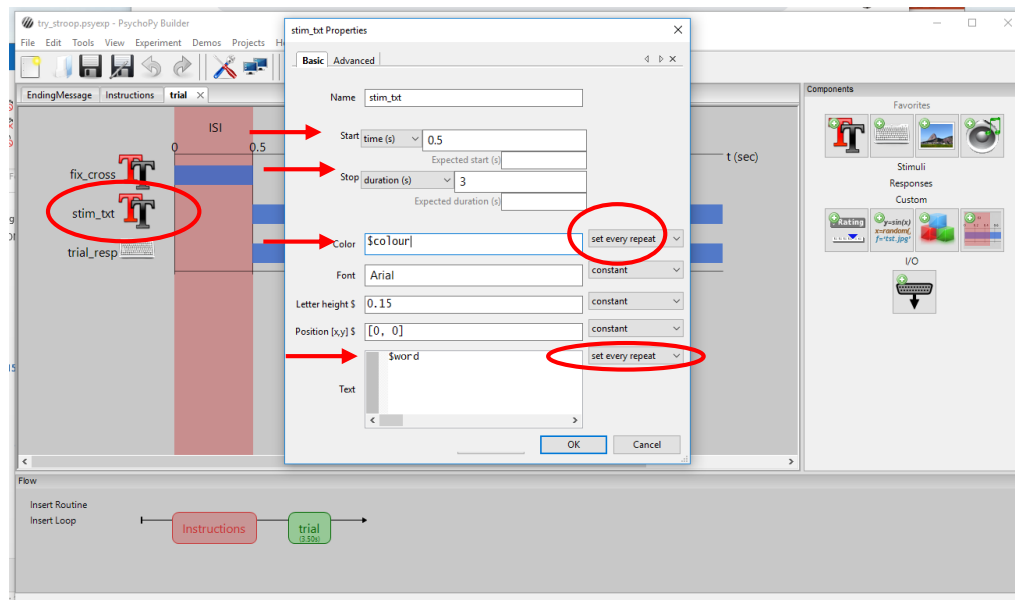


Make sure you are in Instructions routine and click open the 'fix_cross' component

Update the fields in the Text (fix_cross) properties like what is shown on the right accordingly



- Next, with the 'stim_txt' (text) component for presenting a stimulus,
- Update the fields in Properties with settings shown as follows.



Name: stim_txt

Start/time(s): 0.5 (we want the stimulus to appear right after the fixation)

Stop / duration(s) – change it to 3 (you want each stimulus to be presented for 3sec)

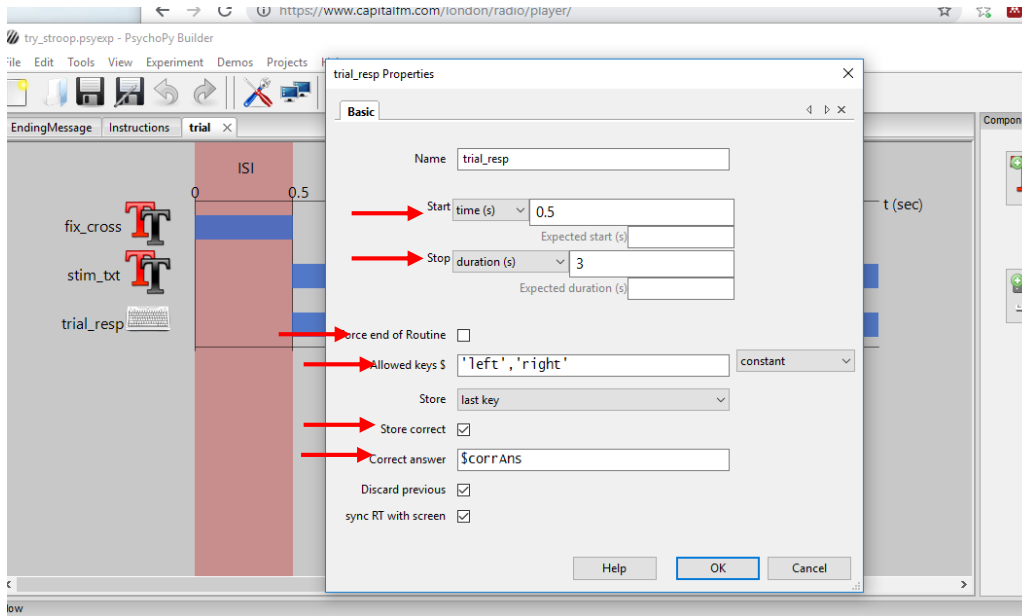
Color: \$colour ('\$' sign to indicate that this information will be taken from the column named 'colour' in the conditions file we are going to link this experiment to)

Text: \$word (similarly, this information will be taken from the linked conditions file, i.e. the 'word' column)

****Finally, for both 'Color' and 'Text' fields, set the drop-down windows next to them to 'set every repeat' instead of 'constant'. (This is because unlike in 'Instructions routine', these fields here in 'trials routine' present info. that changes trial after trial, i.e. they are variables)**

You do not need to change anything else
Click 'OK'

To collect participant's response (to the stimulus), we use a 'keyboard' component



Update the following fields in Keyboard properties:

- **Name:** trial_resp
- **Start/time(s):** 0.5 (so it will only start collecting data at the same time of the onset of the stimulus)
- **Stop/duration(s):** set it to '3' sec
- **Force end of Routine:** ticked if you want the 'trial' routine to disappear once a response is made
- **Allowed keys:** 'left', 'right'
- **Store:** last key (we want to save the participant's final response)

Store correct: tick this field so that in the datafile to be generated, there will be a column showing whether the participant's response matches with our expected 'correct response'

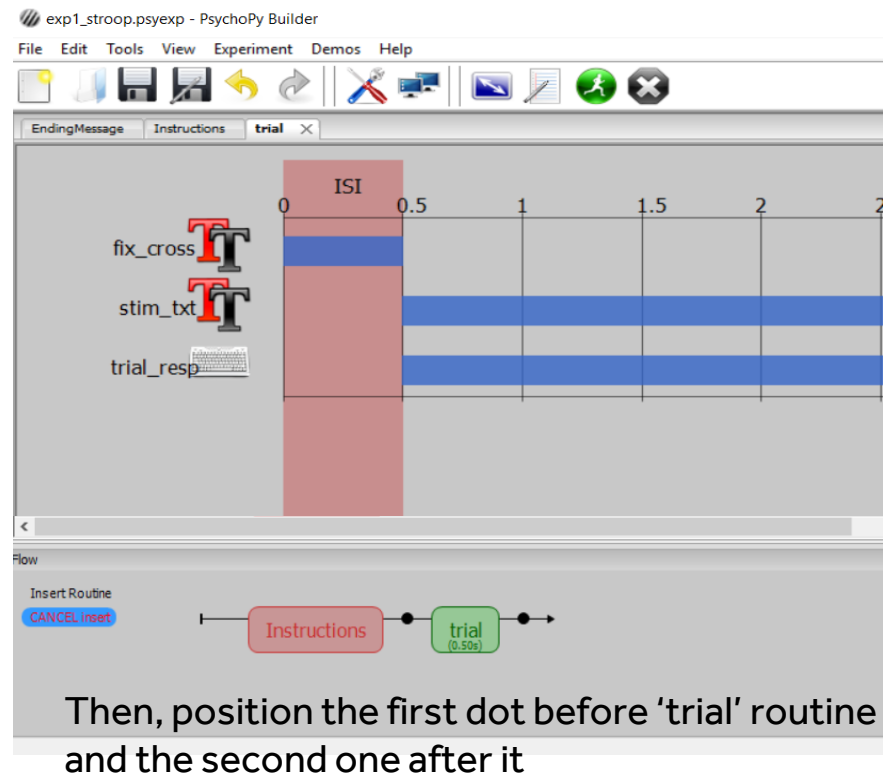
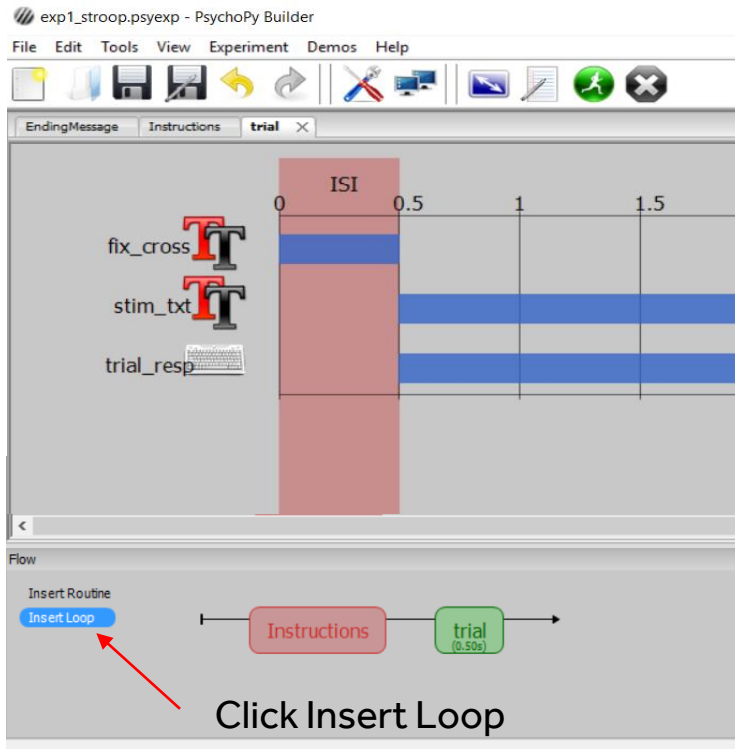
Correct answer: \$corrAns (to indicate that the correct answers are to be taken from the column named 'corrAns in our linked conditions file)

Leave the other settings as they are
Then, click OK to make the changes

LOOPS

In the last few slides, we prepared the 'template' that would be used across trials for stimulus presentation

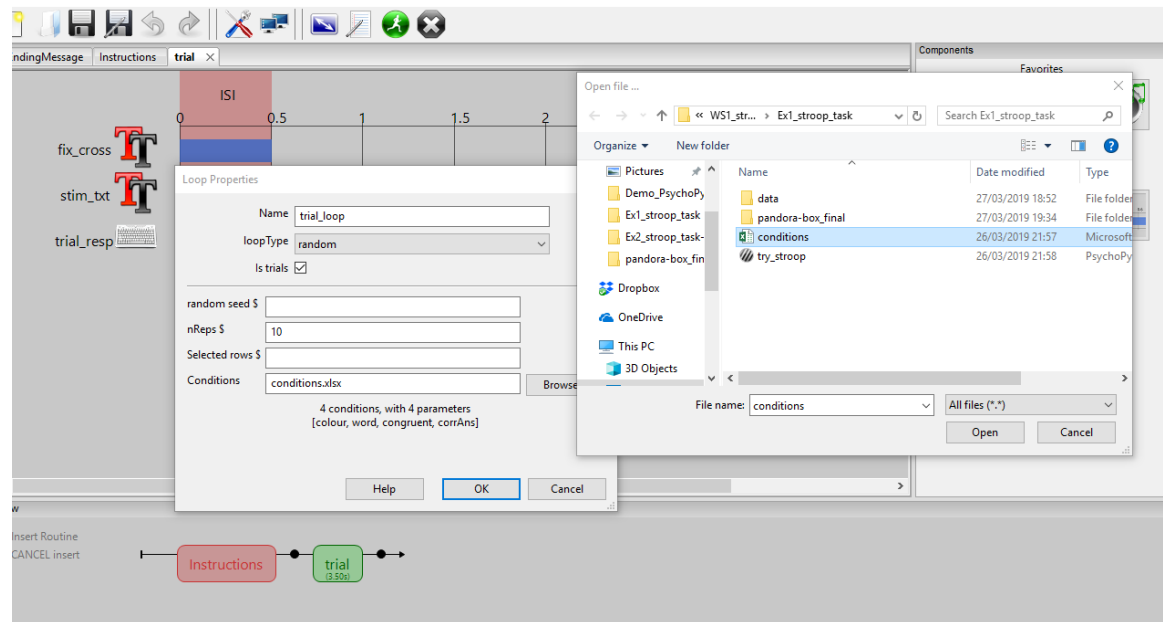
Here, we use the Loop function to control the repetition of the 'template' (i.e. the 'trial' routine)

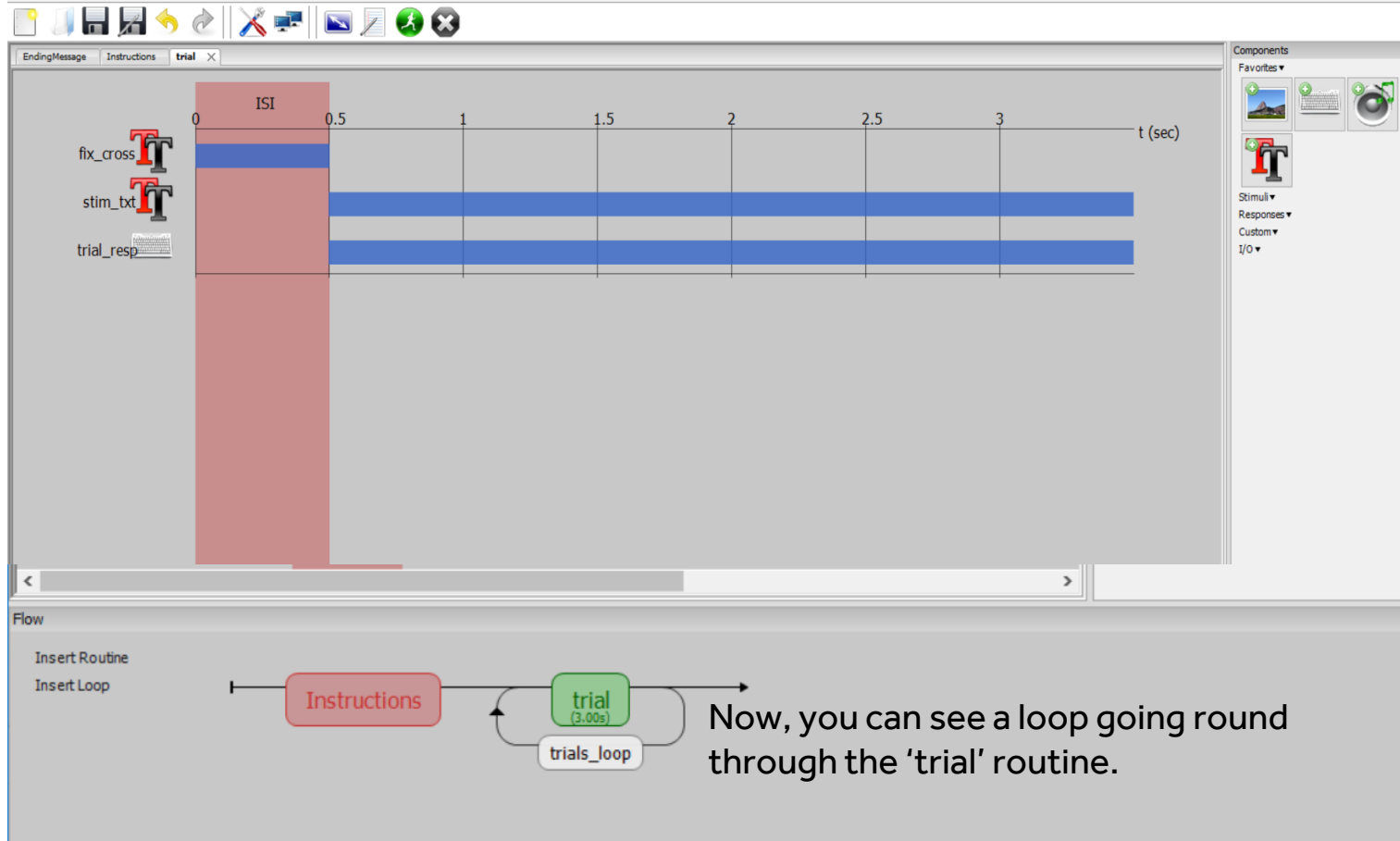


LOOPS

– LINKING IT TO THE CONDITIONS FILE


- In Loop Properties, update the following fields
- **Name:** trial_loop
- **loopType:** keep the 'random' option (so that stimuli will be presented in a random order)
- **Is trial:** keep it ticked to indicate that the routine(s) enclosed by this loop constitute(s) the trials
- **nReps:** put '10' so that the list of stimuli (in our conditions file) will be repeated by 10 times
- **Conditions:** browse to select the conditions file we created earlier





ENDING MESSAGES

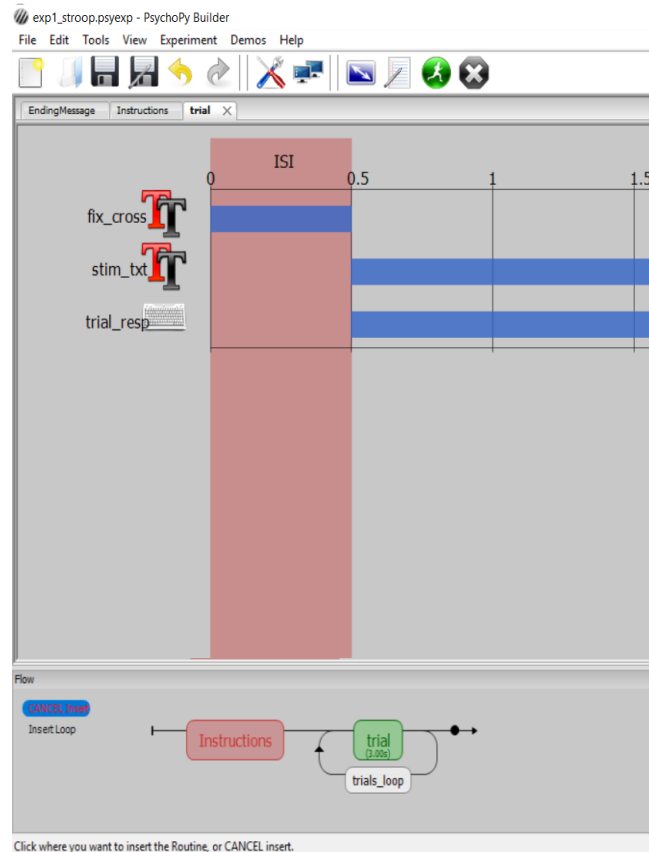
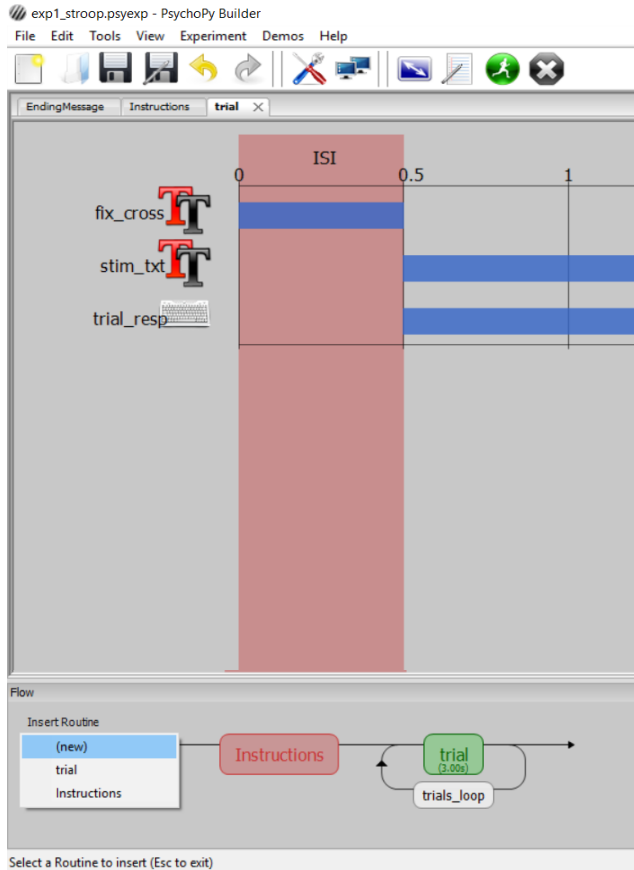
(If you are running of time and we have to move onto Task2, the following slides are the bits you can skip for now.



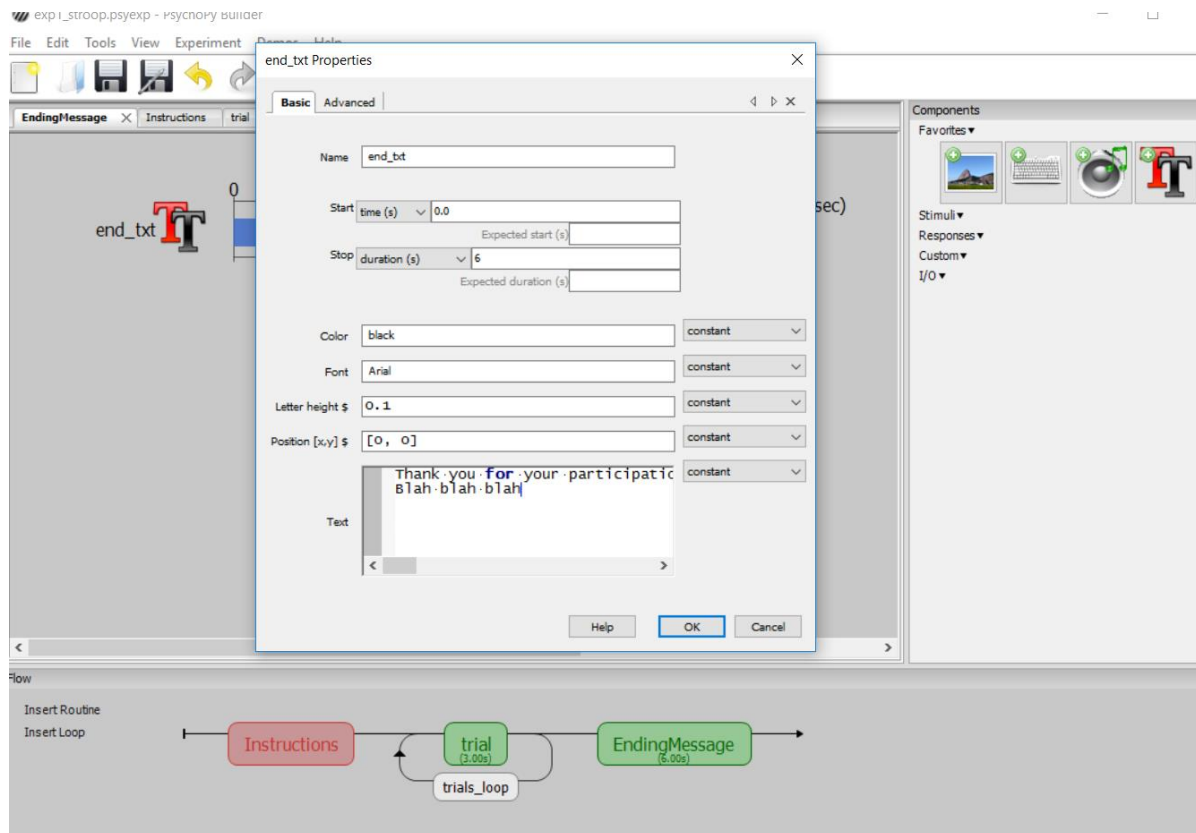
Thank you for your participation!
Blah blah blah

ENDING MESSAGES

At this point, you should be able to create this part on your own.
This is more or less similar to the Instructions routine



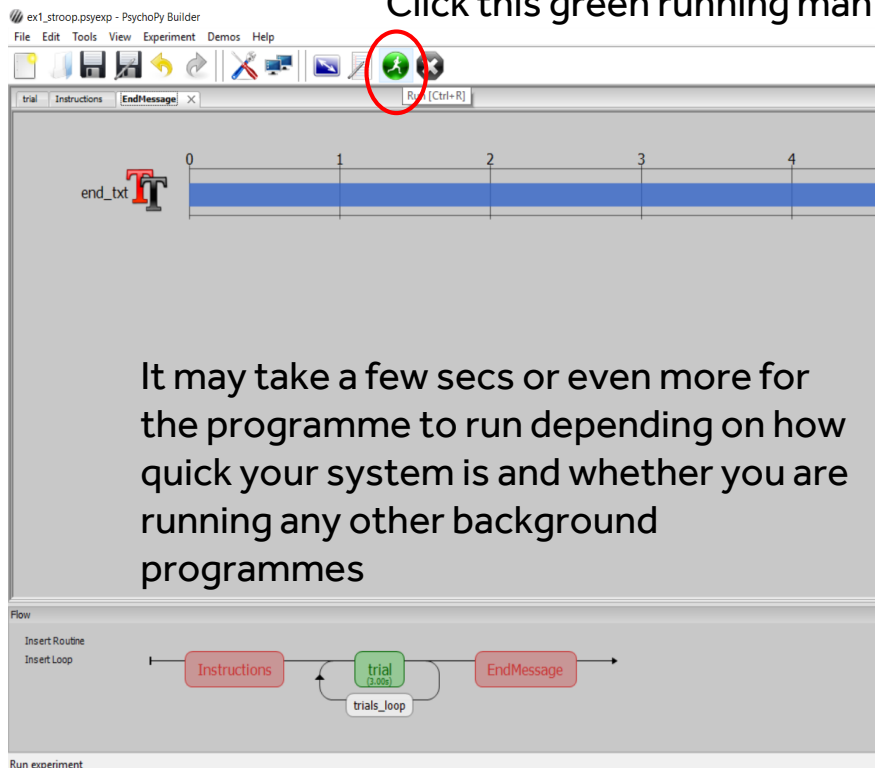
- The core part of the experiment has been created now.
- Let's finish it with some ending messages.
- Similar to previous procedures to create a new routine,
 - click '**Insert Routine**'
 - '**(new)**'
 - Type '**EndingMessage**'
 - position the dot after 'trial' routine (outside of the loop). WHY?



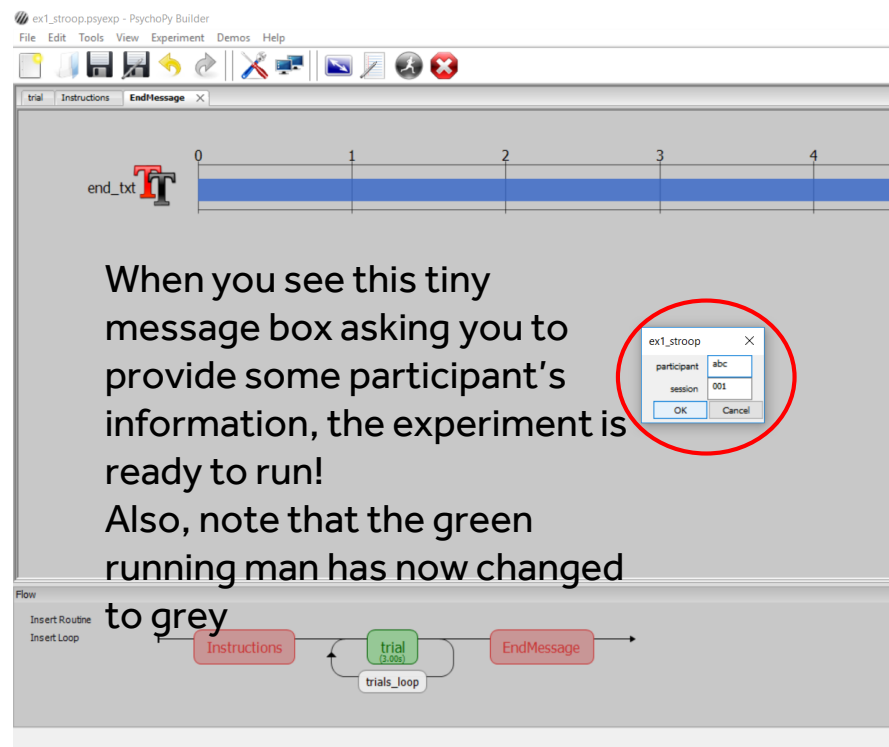
- Use 'Text' component to create ending messages
- Update the fields in Text Properties like what are shown on the left
- This time we set '6' sec to the 'Stop/duration(s)' field so that the Ending Message will disappear at the end of a 6-sec duration.
- We do not need a Keyboard component this time

NOW TEST YOUR PROGRAMME!

Click this green running man



It may take a few secs or even more for the programme to run depending on how quick your system is and whether you are running any other background programmes



When you see this tiny message box asking you to provide some participant's information, the experiment is ready to run!

Also, note that the green running man has now changed to grey

OBTAINING ADDITIONAL PARTICIPANT'S INFO

If you want to collect more participant's information (e.g. gender, age etc) at the beginning of the experiment, you can do this by adding extra fields in Experiment Settings

