

This guide is not meant to be an introduction to PsychoPy (although I do include links to good resources at the start). Instead, it walks through how I built my most recent experiment which had subjects make decisions about sandwiches! The experiment can be found in the file Exp, and is called SandwichStudy.psyexp, and I recommend having that experiment open when reading through this guide. I think the main intention of this guide is to hopefully give some tips and tricks to beginner/intermediate users of PsychoPy, as well as to have a document to reference for my guide on hosting experiments on pavlovia.org.

Feel free to leave any comments about parts of this guide that are confusing, or if you have suggestions of better ways to build parts of this experiment, and I will do my best to keep updating this guide based on these comments.

My ideas and opinions here are just my own, and only reflect my approach to using PsychoPy, and of course, all credit for the amazing PsychoPy software goes to:

Peirce, J. W., Gray, J. R., Simpson, S., MacAskill, M. R., Höchenberger, R., Sogo, H., Kastman, E., Lindeløv, J. (2019). PsychoPy2: experiments in behavior made easy. Behavior Research Methods. 10.3758/s13428-018-01193-y

Table of Contents:

Good Resources for Learning PsychoPy	3
Installation:	3
Overview of SandwichStudy.psyexp	3
Experiment Settings:	4
Experiment Info:	4
PsychoPy Version:	4
Experiment Data	4
Screen Color:	4
Online Settings:	5
Condition Files:	5
Routine and Component Walkthrough:	5
Study_Info_Sheet routine:	6
TextHeader	6
BeginExpCode	6
Info_Cont	6
Full_Info_Sheet / Instructions/ Instructions2	6
VegQuestionLoop	7
VegQuestion Routine:	7
VeggieQuestion Component:	7
VegConfirm Routine:	7
ConfirmCode Component (breaking a loop):	7
trials loop:	8
Initialize Routine:	8
PresentSandMain Routine and Iter Loop:	9
Iter Loop:	9

PresentSandMain Routine:	9
Text Components:	9
TrialCode:	9
MidwayBreak Routine and BreakConditional Loop (Using Loops as a way to conditionally show certain routines)	10
Rating Loop, Initialize Rating Routine, and IngredientRating Routine:	10

Good Resources for Learning PsychoPy

This guide is not going to go into a lot of detail about how to build experiments in general, and is instead just a documentation/walkthrough of my most recently built experiment. For better beginner resources I recommend:

PsychoPy website: <https://www.psychopy.org/documentation.html>

A nice youtube playlist showing how to build lots of simple experiments:

<https://www.youtube.com/playlist?list=PLFB5A1BE51964D587>

This extensive guide to all the components in the PsychoPy builder (I use this for reference all the time):

<https://www.psychopy.org/builder/index.html#builder>

Installation:

If you have not done so already, a good start would be to download the latest version of PsychoPy at

<https://www.psychopy.org/download.html>

At the time of writing this guide, the newest version is PsychoPy 2020.1.3

Overview of SandwichStudy.psyexp

The study I am documenting involved subjects making decisions about which of 2 sandwiches they would prefer to order. The goal of this experiment was to try and measure preferences for the individual ingredients on each sandwich. The main trials looked like this:

Which of these two sandwiches would you rather order?
Press 'space' to see more ingredients.
Press 'A' for Sandwich 1, press 'L' for Sandwich 2

Sandwich 1	or	Sandwich 2
<hr/>		<hr/>
Meatballs		Chicken

Trial: 38 of 80

At the start of each trial subjects saw just the first ingredient, and pressed space to reveal more ingredients (with a .8 /s delay between when they could hit space again). They used the keyboard to choose between sandwiches and could make their decision at any time in the trial. The other type of trial we had was asking subjects to rate their preference for different ingredients on a scale from 1-7, and this happened at the end of the experiment.

The high-level structure of the experiment went as follows:

Consent form and Instructions -> Asking whether a subject was vegetarian or not -> 80 of the main sandwich decision trials (shown above) with a break after 40 trials -> 10 ingredient rating questions -> End of experiment screen.

Experiment Settings:

While the default PsychoPy experiment settings are a great starting point, I made a few changes to these settings that I will document here. You can modify these settings by opening the experiment settings tab (the icon at the top of the builder screen with a small gear on it).

Experiment Info:

I deleted the default setting to record a session, and instead just have participant info. You can add or remove as many inputs of experiment info as you would like, and when you run your experiment a window will open asking you to input values for these different experiment info fields before starting the experiment. When running studies online you can automatically pass in this information (such as participant id) from another website like sona or prolific so that subjects do not have to manually fill this in.

PsychoPy Version:

As noted in the ReadMe, I changed my running version of PsychoPy to 3.2, because this is when I originally built this experiment, but normally this will be set to the most recent version of PsychoPy you have installed. Good to know that you can change this to older versions though.

Experiment Data

In the “data” tab in experiment settings, you can customize how you want the experiment data saved. You can adjust the format for the file names. I basically leave this as the default, although if you made edits to the experiment info you might have to adjust what info you want saved in the file name. I also just save the trial by trial csv data since I like having basically every bit of recorded data available, but you can play around with how you would like your data saved to find what suits you. To see an example of how this trial by trial data looks on this experiment, open up the sub folder called data, and there should be a csv file labelled Demo Data_trial.csv that you can explore.

Screen Color:

I really do not like the default grey background of PsychoPy studies. So in the “screen” tab in experimental settings, you can modify the color of the screen. You can do this by right-clicking on the field “color”, and then selecting your background color (I went with white). Note, that if you go with white, you need to change the default color of any text components, because this is also white. More information on how to best handle this in [BeginExpCode](#).

Online Settings:

In the online tab in experiment settings, there are two fields you can customize, completed url and incompleted url. As you would expect this means that if a subject leaves/quits the study early they get

redirect to incomplete url, and if they finish completely they get directed to completed url. This is really important for automatically granting participation credit in Sona or Prolific, and you can read more about that in my hosting PsychoPy experiments online guide which I will **hopefully remember to link to**. I wouldn't advise editing the Output Path, or Export html settings, since I don't really know how those work, and the defaults worked just fine for me.

Condition Files:

This experiment uses 4 external condition files which are .csv files. If you do not know how condition files are used in loops in PsychoPy see <https://www.psychopy.org/builder/flow.html#loops>

There are basically two types of condition files I use:

The ones with the sandwiches for the main trials (CleanDataMoreIng.csv and CleanDataVeggie) which contain the ingredients of these sandwiches for the regular and vegetarian sandwiches respectively. Each row in this file is all the information used to present the stimuli for a single trial.

These files have the exact same column or variable names, which involve the 8 ingredients on the first sandwich (s11-s18), the 8 ingredients on the second sandwich (s21-s28), and then 3 other variables, the maximum number of ingredients for both sandwiches, and then the number of ingredients on sandwich 1, and sandwich2, these variables for number of ingredients are important for later in the code.

The other condition files are Top10IngsLong.csv and Top10IngsVeggie.csv, and again these have the exact same column or variable name, which is just ingredientforslider, and as you can see in the rows, these are just the names of the 10 most common ingredients across the regular and vegetarian sandwiches respectively.

Routine and Component Walkthrough:

Each of the following sections will cover a routine in the experiment SandwichStudy.psyexp. A routine is shown in the bottom tab of the PsychoPy builder window labeled "flow". This flow documents the order of routines in your experiment, and contains any loops around experiments (If you are unfamiliar with flow, routines, loops, or components I advise starting with some of the beginner resources listed above). Each of these sections will also cover the specific components, loops, or code that make up each routine. A component is a specific stimulus or response option on a given routine, such as any text components, keyboard/response/rating scale components, and even your custom code snippets. When you select a routine from the flow section of the builder interface, all of the components of that routine are shown in the main builder window.

Study_Info_Sheet routine:

This routine is the very first screen a subject sees, and contains three components.

TextHeader

This is just a text component that contains the text that appears on screen for this trial.

BeginExpCode

This code initializes a lot of custom variables that are used throughout the rest of the experiment. Most of them adjust the location, size, color, text wrap, or other details about text components used later in the experiment. I choose to create these variables so that when I am tweaking the final look of this experiment, if I decide to change the text size of some components, I can just change this once in this section of code rather than having to manually go in and change it in all the components that depend on this.

Because I changed the background color of my experiment to white, I simply created a variable in this section called `TextColor= [-1.000,-1.000,-1.000]`, which is the value in PsychoPy for black, and then in all other text components, I set color to `$TextColor`.

I also created a random sequence of 1s and 0s which controls the order of sandwiches in the main trial. This required different code for python and JS (although I believe with the new auto-translate feature you can use numpy in JS as well), and if you look at the end of the experiment tab it demonstrates how to save a custom variable to the csv file output at the end of the experiment.

Info_Cont

This is a keyboard component that the subject uses to advance to the next routine. I like to use keyboard components because you can limit the response to a single key which reduces the chance that someone will accidentally hit a key or mouse and skip past a routine.

Full_Info_Sheet / Instructions/ Instructions2

These routines contain the study information sheet required by my IRB, and then Instructions and Instructions2 contain text components with the specific task instructions for this experiment. All of these routines use a keyboard component to allow the subjects to move to the next routine when finished reading. If you want to put an extra measure in place to try to reduce subjects just clicking through, I recommend putting a 5 to 10 second delay on the start of the keyboard component to encourage subjects to at least start reading the instructions.

VegQuestionLoop

This loop and the two routines within it handle asking whether a subject was a vegetarian, letting them confirm or change their answer, and the setting the different variables for later in the experiment so that they were only shown sandwiches with vegetarian ingredients. The way this loop works is that it will continue for up to 100 iterations (the variable `VegQuestionLoop` is set in `BeginExpCode` to 100) if the subject continues to not confirm their response, or will break when the subject confirms their response of vegetarian or not vegetarian (see below).

VegQuestion Routine:

This routine asks subjects whether they are vegetarian or not, and has them respond using a keyboard component by selecting 'y' for vegetarian, or 'n' for not vegetarian.

VeggieQuestion Component:

This code component creates a new variable ConfirmText which changes based on whether the person initially said they were or were not a vegetarian. This variable is then used in the next routine

VegConfirm Routine:

This routine is where subjects confirm their choice on the previous routine using “P” to confirm, or “Q” if they would like to start the VegQuestionLoop over, and change their original selection.

ConfirmCode Component (breaking a loop):

The code in this component demonstrates how to use a series conditional statements and previous responses to break out of a loop. This code breaks out of the loop if the response to the keyboard component is “P”. The method for breaking out of loops depends on the version of PsychoPy you are using, so if you do not know how to do this for your code see:

<https://discourse.psychopy.org/t/loop-finished-true-no-longer-working/11190/3>

When the response is “P”, it also defines some variables that are used later in the experiment, ConditionFile and ConditionFile2. These variables take the name of csv condition files stored in the same folder as the PsychoPy experiment that contains the information used in the sandwich decision trials and ingredient rating trials.

If they are vegetarian (responded ‘y’ on the previous routine), ConditionFile and ConditionFile2 are defined as the vegetarian csv files.

If they are not vegetarian (responded ‘n’ on the previous routine), ConditionFile and ConditionFile2 are defined as the csv files containing the regular sandwiches. For more information about how to use condition files (they are very important) see: <https://www.psychopy.org/builder/flow.html#loops>

After the definition of these condition files is completed, the final line of this code breaks out of the current VegQuestionLoop.

trials loop:

The loop called trials, and the 3 routines within it contain the routines that are involved in the main sandwich decision trials.

Clicking on trials in the flow window shows that the conditions option is set to a variable \$ConditionsFile. This was defined in the previous routine [ConfirmCode Component](#). This contains the data on all the sandwiches that subject will choose between. The loop type is set to random which means it will randomly order the rows of this conditions file, and each row represents a single trial.

The initialize routine involves only a code component that pre-builds the stimuli for each sandwich trial. The PresentSandMain routine is where the sandwiches are actually shown, and the decision is recorded, and the Iter loop handles presenting a new ingredient when the subject presses ‘space’ and ends when they select a sandwich using ‘a’ or ‘l’.

The MidwayBreak routine and BreakConditional loop handle showing the break screen, and the loop serves as a conditional that only shows the break screen when the subject has completed half (40) of the trials.

Initialize Routine:

This routine involves a single code component SetParams, I'll walk through what happens in this code component line by line.

- First, the variable MaxIngs is set to the current value of IngsInRow from the Conditions file. This variable ends up dictating how many times the subject can hit space for more ingredients before they are told all ingredients are presented

- Some other variables are defined such as count=0 which is incremented by 1 whenever the subject hits space to see more ingredients on the following routine. Also TrialCount is incremented by 1 each time through this routine, and as you would expect this is counting the trial number. Finally TakeBreak is set to 0, and this is what is used later to decide whether to show the break screen.

- Next there is a conditional statement, where if TrialCount is equal to half of the total number of trials (which is 40), then TakeBreak is set to 1, this is used later to decide whether to show the break routine.

- Next a number of strings are defined. I'm not going to go into extensive detail about what is going on here, but it is essentially building the set of ingredients that will be shown to the subject when they start the next routine, after they hit the spacebar once, after they hit the spacebar twice, etc. If you are specifically interested in showing a text component that "grows" based on user input **I will hopefully add an appendix about that soon.**

- Finally another string is defined, CountText, which is the string used to present the trial count to the subject in the bottom right of the actual trial screen.

PresentSandMain Routine and Iter Loop:

This is the main routine and loop where subjects see the sandwich ingredients and make their decision.

Iter Loop:

The Iter loop has no condition files, and repeats 1000 (an arbitrarily large) number of times, but in practice is broken by subject's response in the routine it is around, PresentSandMain. This is not normally how loops are used in PsychoPy as they normally contain a conditions file, and loop around individual trials, but I needed this loop to keep updating the text shown on the screen when a subject hit space to see another ingredient.

PresentSandMain Routine:

This Routine contains a large number of components, most of which are just text components, and the bulk of the action is in the code component TrialCode. There is a keyboard component that accepts space, L, and A, which has a .8s start delay, which subjects use to reveal new ingredients or choose between sandwiches.

Text Components:

The constant text components include Sandwich1, Sandwich2, or, Instr. These include the Sandwich labels, the or in the center of the screen, and the main trial instructions at the top of the screen.

The variable text components include Ingredients1, Ingredients2, InstructionEndOfTrial, and TrialCounter. Respectively, these include the strings that present the ingredients for sandwich 1 and sandwich 2, the red instructions saying “All ingredients presented”, and the trial counter at the bottom left of the screen. All of the text in these components is defined in the code components either in this trial, TrialCode (for Ingredients1, Ingredients2, and InstructionsEndOfTrial), or in the Initialize routine (for TrialCounter).

TrialCode:

TrialCode involves the code that really drives this trial, described line by line below:

- First off the variable Count is incremented by 1, this is used to indicate what ingredient should be presented, so it is incremented every time this trial is reiterated over (which only happens when the subject presses space to reveal a new ingredient).
- The next series of conditional statements defines what ingredients will be shown to the subject for the trials on each iteration. The conditional involving Flip just handles when the sandwich 1 in the conditions file shows up as sandwich 1 in the experiment, or vice versa. The nested conditionals select the format of the sandwich shown from the large array of strings defined in the Initialize routine. I subtract 1 from the indexing of this array since I increment count before doing this indexing, and Python uses 0 indexing. Essentially these conditionals compare the current value of Count to the number of ingredients in both sandwiches. If Count is less than or equal to this number of ingredients it selects the string from the SandwichTextX array that has the number of ingredients shown equal to Count (in other words when Count=1, it shows the selection from this array that only has the first ingredient shown, if they hit space again and Count=2 then it selects from this array the string that has 2 ingredients shown, and so on). But if Count is greater than the number of ingredients in that sandwich, then it continues to just select the string that is formatted based on the max ingredients in that sandwich. It does this for each sandwich separately since the sandwiches don't have to have equal number of ingredients. **For more details on this string array process see this appendix that I have not yet written.**
- Finally, there is a conditional statement at the end for when Count is greater than or equal to the maximum number of ingredients on that trail, that defines the variable used in InstructionEndOfTrial component once all ingredients have been presented.

MidwayBreak Routine and BreakConditional Loop (Using Loops as a way to conditionally show certain routines)

The MidwayBreak is a very standard routine that has some text explaining that this is a break, lasts for 3 minutes, and accepts a keyboard response if the subject wants to end the break early.

The more interesting feature is in using the BreakConditional loop as a way to only show the break at the halfway point in the experiment. In the [Initialize Routine](#), I define a variable TakeBreak, that is usually set to 0, but when the trial count=40 is defined as 1. Inside the details of the BreakConditional Loop, you can see I set the detail nReps to \$TakeBreak, and this has the effect of only engaging in this loop and showing this routine when TakeBreak is non zero. This method of defining variables and then using them inside the loop details is a powerful way to only show some components of your flow conditionally, whether it is based on trail count, or past responses, or any other previously defined variables you can use.

Rating Loop, Initialize Rating Routine, and IngredientRating Routine:

The rating loop involves the subject making the 10 preference self-reports about the 10 most common ingredients on the sandwiches they saw, indicated by ConditionsFile2 which is defined based on whether they are vegetarian or not.

There is nothing that interesting about these routines in general that was not discussed previously, besides the slider component which is rather self explanatory if you open up the slider details and hover over each field, which gives you a good description of what changes to those fields actually do.

Goodbye!

Thanks for giving this a read, feel free to leave any comments about anything that is confusing and I will try to provide further explanations, details, and revisions as I am able.

Good Luck with your experiment building!