

# Lab 1 - PsychoPy Basics

## Instructions Routine

We will need two different types of components for our instructions routine: the **Text and Keyboard components**.

### Text

- Name: the variable name that you want to give to Text object
- Start: the onset time that the text will be presented
- Stop: determines the offset time (can be specified in durations or times)
- Text: contains the text that you want to display
- Letter height: the "font size" of the text
- Refer to this for more details: <https://www.psychopy.org/builder/components/text.html>

In general, getting the aesthetics of the text component to display nicely takes some trial and error. I usually tinker around with the Layout and Format panels to get it right.

### Keyboard

- Name: the variable name that you want to give to object
- Start: the onset time that the object will be presented
- Stop: determines the offset time (can be specified in durations or times)
- Allowed keys: specify which buttons on the keyboard you want to listen to.
- Store: If more than 1 response is registered, which key do you want to store? last key, first key, all keys, nothing
- Refer to this for more details: <https://www.psychopy.org/builder/components/keyboard.html>

For instructions, we often set the Keyboard duration to indefinite by leaving the "Stop" field empty. This gives subjects the opportunity to take as much time as they need to read the instructions. Make sure "Force end of Routine" is checked if you want a Keyboard response to advance to the next routine.

### Steps:

1. Create an `instr` routine.
2. Add an `instr_text` Text object and a `instr_kb` Keyboard object to the routine
3. Copy and paste the instruction text below to the `instr_text` object.
4. Delete the value of the stop duration of `instr_text`.
5. In `instr_kb`, make sure "Force end of Routine is checked". Set "Allowed keys" to 'space'.
6. Finally, test and tinker with the parameters

### Instruction Text (to copy to your experiment):

Welcome to the study!

In this experiment, an array of images will be presented on the screen for 3 seconds.

After a 5-second delay, you will be shown a series of individual images, some that were presented in the array and others that weren't. Your task is to indicate if the image is old or new by responding "f" if it is "old" and "j" if it is "new".

When you're ready, please press the spacebar to begin the experiment.

## Trial Routine

We will now get into the meat of the experiment. In this routine, we will 1) present an array of images, 2) show a delay screen, and 3) show a probe image to test their old/new recognition memory.

### Image

- Name: the variable name that you want to give to object
- Start: the onset time that the object will be presented
- Stop: determines the offset time (can be specified in durations or times)
- Image: relative file path to the image (relative meaning relative to .psyexp file)
- Size: controls the width and height of the image
- Refer to this for more details: <https://www.psychopy.org/builder/components/image.html>

If you want to assign a different image to be presented depending on the trial, the drop-down menu next to the "Image field" needs to change to "Set every repeat". We will go over this use case today.

### Loop

- Name: the variable name that you want to give to object
- nReps: number of repeats for this loop
- Refer to this for more details: <https://www.psychopy.org/builder/flow.html>

Loops can be applied around a single routine or multiple routine to enable iteration in your experiments. Experiments are frequently divided into blocks, which are in turn comprised of trials. Trial loops can be made by looping around a routine. Block loops can be made by looping around trial loops. We will employ just a trial-level loop today.

Please note that there are a lot more fields in the Loop component that I won't go over today. There are many ways to use loops and to iterate through trials in your experiment, but we won't cover that today.

### Custom Code

- Before Experiment: Things that need to be done just once, like importing a supporting module, which do not need the experiment window to exist yet.
- Begin Experiment: Things that need to be done just once, like initialising a variable for later use, which may need to refer to the experiment window.
- Begin Routine: Certain things might need to be done at the start of a Routine e.g. at the beginning of each trial you might decide which side a stimulus will appear.
- Each Frame: Things that need to be updated constantly, throughout the experiment. Note that these will be executed exactly once per video frame (on the order of every 10ms), to give dynamic displays. Static displays do not need to be updated every frame.
- End Routine: At the end of the Routine (e.g. the trial) you may need to do additional things, like checking if the participant got the right answer
- End Experiment: Use this for things like saving data to disk, presenting a graph(?), or resetting hardware to its original state.
- Copied from this page: <https://www.psychopy.org/builder/components/code.html>

For the most part, I write code in the "Begin Experiment" and "Begin Routine" panels. The "End Routine" panels is often used to reset/update variables or to write trial data to a variable.

When you create new variables in your code components, these can be passed as values in the field boxes of the components if you use the `$` character followed by the name of the variable. This is useful if you want to programmatically assign the number of repetitions in a loop or the image file of a particular trial. We will go over both these cases today.

The Code component allows you to integrate custom code to your experiment, allowing for more robust control of your components.

## Steps

1. Create a new routine called `task`.
2. Add 4 new Image components called `array_im1`, `array_im2`, etc. Change the layout properties so that the 4 images are presented in a 2x2 matrix. Set the durations to 3s.
3. Add a Text component called `delay`. Change text to a "+". Change start time to 3s, and duration to 5s.
4. Add 1 new Image component called `test_im`. Change start time to 8s and delete the value in duration so that it persists infinitely.
5. Add 1 Keyboard component called `key_resp`. Change start time to 8s, allowable keys to 'f' and 'j', and delete duration value.
6. Insert loop around the `task` routine. Name the loop `trials` and specify 10 repeats.
7. For all the images, in the Image field, add the variable name of the image that will be used for that Image component (i.e., the `array_im1` component will be `$array_im1_im`). *Make sure "set every repeat" is selected rather than "constant".*
8. Create a code component. The following code chunks will go in this component. I will talk it through in the live demo.

In the Begin Experiment panel:

```
In [ ]: # 10 trials
# should this be an old or new trial?
trial_oldnew = [1, 1, 1, 1, 1, 0, 0, 0, 0, 0] # 1 will be a new trial, 0 will be old
shuffle(trial_oldnew)

# assign sequence of images array_im1, array_im2, etc
im_nums = list(range(1,51))
array_im1_seq = list()
array_im2_seq = list()
array_im3_seq = list()
array_im4_seq = list()
test_im_seq = list()

for i in range(10):
    # shuffle available list of images (numbered 1 to 50)
    shuffle(im_nums)

    # assign array ims
    array_im1_seq.append(im_nums[0])
    array_im2_seq.append(im_nums[1])
    array_im3_seq.append(im_nums[2])
    array_im4_seq.append(im_nums[3])

    # assign probe im depending on whether the trial is old or new
    if trial_oldnew[i] == 0:
        test_im_seq.append(im_nums[4]) # grabs one of the ims not in the array
    elif trial_oldnew[i] == 1:
```

```
array_indices = list(range(3))
shuffle(array_indices)
test_im_seq.append(im_nums[array_indices[0]])
```

```
# initiate trial counter
this_loop_N = 0
```

In the Begin Routine panel:

```
In [ ]: # assign array_im1 image, etc
array_im1_im = 'resources\Animal_' + str(array_im1_seq[this_loop_N]) + '.jpg'
array_im2_im = 'resources\Animal_' + str(array_im2_seq[this_loop_N]) + '.jpg'
array_im3_im = 'resources\Animal_' + str(array_im3_seq[this_loop_N]) + '.jpg'
array_im4_im = 'resources\Animal_' + str(array_im4_seq[this_loop_N]) + '.jpg'

# assign test_im_im
test_im_im = 'resources\Animal_' + str(test_im_seq[this_loop_N]) + '.jpg'
```

In End Routine panel:

```
In [ ]: # update trial counter
this_loop_N += 1
```

And that's it!! You have your first PsychoPy/Python experiment!