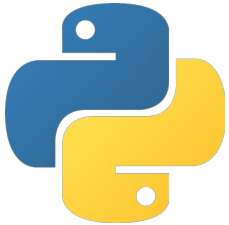

Workshop: **Introduction to Python**

Experiment Creation

Christian C. Luhmann
Stony Brook University

Relevant Packages

- **psychopy**
- pyserial
- pyparallel
- pyopengl
- pyglet
- moviepy
- pillow

PsychoPy

- Originally created by Jon Pierce
- Initiated as a python replacement for Psychtoolbox
  - But it has grown in to much, much more
- It is recommended to install the Psychopy **standalone** alongside any data-centric python installation you might have (e.g., Anaconda)
- The standalone version of PsychoPy includes...
  - Python (2.x or, as of April 2018, 3.x)
  - PsychoPy
  - all the other packages required by PsychoPy

PsychoPy

- Coder
  - Coder is PsychoPy's IDE (like Spyder)
  - Provides already-written demos and examples
- Builder
  - No programming required
  - Experiments are built using a graphical interface
  - Demos/examples already built
- Can also use PsychoPy as a regular package within python (2.x or 3.x)

PsychoPy

- Running PsychoPy (using the shortcut created during installation) should get this…

Builder

- Routines
  - Describe timing of stimuli, instructions, responses, etc.
  - A set of events that will always occur together in a fixed order

- Flow
  - Control the way in with the Routines are combined, repeated, and controlled
  - Like a flowchart

Builder

Let’s use Builder to **build** some experiments!

Shooter Task – Builder Version

Specification:

- Show a page of instructions
- Run 1 trial per image:
  - Show an image
  - Collect a response
  - Show a blank screen (ITI)
- Tell the subject that they’re done

Shooter Task – Experiment Settings

- File->New

- Experiment Settings (blue button to the right of Monitor Center button)
  - Full-screen window: unchecked
  - Window size (pixels): [900,900]
  - Can leave the rest as they are

Shooter Task – Instructions

- Experiment->New Routine(Shift+Ctrl+N)
  - Name: instructions
- Click on the Text component button (to the right)
  - name: instructText
  - duration: leave blank
  - color: black
  - text: Whatever you want your instructions to say, but ask for a keypress
- Click on the Keyboard component button (to the right)
  - name: instructResp
  - allowedKeys: leave blank
  - store: nothing
  - forceEndTrial: checked
  - storeResponseTime: unchecked

Shooter Task – ITI

- Experiment->New Routine(Shift+Ctrl+N)
  - Name: ITI
- Click on the Text component button (to the right)
  - name: itiText
  - duration: 0.5
  - text: leave blank

Shooter Task – End of Task

- Experiment->New Routine
  - Name: thanks
- Click on the Text component button (to the right)
  - name: thanksText
  - duration: 10.0
  - color: black
  - text: Whatever you want your instructions to say, but ask for a keypress
- Click on the Keyboard component button (to the right)
  - name: thanksResp
  - allowedKeys: leave blank
  - store: nothing
  - forceEndTrial: checked
  - duration: 10.0
  - storeResponseTime: unchecked

Shooter Task – Trials

- Experiment->New Routine
  - Name: trial
- Click on the Image component button (to the right)
  - name: trialImg
  - duration: 1.0
  - image: leave blank for now (will change later)
- Click on the Keyboard component button (to the right)
  - name: trialResp
  - allowedKeys: [‘1’, ‘2’]
  - store: last key
  - storeCorrect: checked
  - correctAns: leave blank for now (will change later)
  - forceEndTrial: checked
  - duration: 2.0
  - storeResponseTime: checked

Shooter Task – Flow

- Insert Routine
  - Select “instructions”
  - Click to the left of “trial”
- Insert Routine
  - Select “thanks”
  - Click to the right of “trial”
- Insert Loop
  - name: trials
  - loopType: random
  - nReps: 1
  - trialListFile: trials.csv

trials.csv

- Comma-delimited text file containing:

```
img,correctAns
./images/zaba011.jpg,black
./images/zabu02w4.jpg,black
./images/zawa90d3.jpg,white
```

- First line is the header, telling you what each column is
- Remaining lines describe each trial

trials.csv

- Need to go back and add in the trial info that will change each time through the loop
- In the trial routine, click on the keyboard component we added earlier
  - Change the correctAns to $thisTrial.correctAns
- In the trial routine, click on the image component we added earlier
  - Change the image to $thisTrial.img
  - Change the dropdown box (to the left of image) to “set every repeat”

Asks for the info associated with the “current” line of the trialLoop.csv file. It’s “$thisTrial” because we called our loop “trials”. If we had named the loop “blocks”, we would want “$thisBlock.correctAns” instead.

Asks for the info from the “correctAns” column of the trialLoop.csv file

$thisTrial.correctAns

Same thing here, but now we want to ask for the info from the “img” column.

$thisTrial.img

Coder

Let’s use Coder to **build** that same experiment!

Take-homes

- You've now seen an **experiment created** in psychopy (twice!)
  - Builder
  - Code
- You've seen some of the functionality that **psychopy** provides
  - Window creation and management
  - Stimulus creation and manipulation
  - Response collection
  - Timing
  - Much more (other peripherals, logging, staircasing, etc.)
- Online experiments are available via **Pavlovia.org**

Outline

1. Overview
2. Ways of using Python
3. Python basics
4. Data set overview
5. Data wrangling
6. Statistics
7. Plotting
8. Experiment creation

The End(ish)

- Materials available
  - github.com/cluhmann/python-psych-workshop
- Thank you
- Questions?
  - Feel free to contact me:s
  - christian.luhmann@stonybrook.edu