

**1/17/2019**

**PSYC 734  
Python/Psychopy**

**Spring 2019**

**Time:** 1:30 - 4pm Thursday

**Classroom:** Arch lab conference room 2073, David King Hall

**Instructor:** Dr. Matt Peterson

**Office:** 2058 David King Hall

Ph: 993-4255

email:

(please put PSYC 892 in the subject header)

**Office Hours:** 12:30-1:20 Tuesdays (*tentative*) or by appointment.

I fully encourage you to contact me by email if you have any questions about the class.

**Text:** None. However, there are many great web sites devoted to Python. I will be focusing on Python version 3.

**Software:** I am going to focus on Anaconda *Spyder*, which is a free development environment. At times I will focus on PsychoPy *Coder*. *Coder* is not as nice as *Spyder*, but it is helpful to be familiar with both. *Conda* is a website which provides downloadable packages (language extensions). All work on Mac, Windows, and Linux.

Anaconda Spyder (part of Anaconda Navigator)

<https://www.anaconda.com/download/> Download Python 3.x

PsychoPy

<http://psychopy.org/installation.html> Download 1.9x (as of 10/29/2018)

Conda packages:

<https://anaconda.org>

Useful Python web sites:

<https://www.python.org/about/gettingstarted/>

<http://www.psychopy.org/coder/tutorial1.html>

<https://docs.python.org/2/tutorial/index.html>

<https://www.codecademy.com>

**Prerequisites:** graduate student in Psychology

**Objectives:** In this course you learn to perform data analyses, run simulations, and program psychology experiments in Python. This course will not teach you how to prototype user interfaces (but it would be a good springboard), nor will it teach you to use the *Psychopy Builder* (the GUI interface for non-programmers).

Why Python? Several years ago, I was looking for a language that would allow me to crank out a quick program, without going through all of the hassle required by C++. I found Python to be incredibly easy to learn – I say that it is probably the easiest general-purpose programming language to learn. It's user base is skyrocketing, and there are a plethora of free, open-source, language extensions available (SciPy, NumPy, PsychoPy, etc.). It is especially popular in the sciences and data analytics, and language extensions are available to allow your code to interface with virtually any piece of lab equipment.

**What we will be learning:**

Python language This is the language we will be using for controlling the flow of the program. I will be teaching you to write full-fledged programs, and not simple scripts. I will start by teaching old-school procedural programming, but we will also be covering object-oriented programming.

PsychoPy This is a cross-platform language extension and development environment (Windows and Mac) for writing psychology experiments. It includes functions for precise stimulus timing, keyboard input, etc., and includes built-in functions for interfacing types of lab equipment (e.g. Tobii and Eyelink eyetrackers).

Programming strategies: Knowing the language and the PsychoPy isn't enough: you also need to know *how* to program experiments. That is, designing a program involves a great deal of problem solving in order to come up with the best strategies.

**Types of programs covered (tentative)**

- Data parsing
- Data analysis and graphing
- Statistics
- Montecarlo simulations
- Flanker tasks
- Attentional Blink
- Psychological Refractory Period (PRP)
- Visual search
- Multiple object tracking
- Event Related Potentials
- Eye Tracking

**Attendance Policy:** Since much of this class will involve discussion, I expect (barring unforeseen circumstances, conferences, etc.) to see you in class each week. Skip class at your own peril.

**GMU Honor Code:** George Mason University has a code of Honor that each of you accepts by enrolling as a student. My expectation is that all of the work you do for me in this class will be the work of one individual. Having said that, I fully encourage you to discuss the readings and topics raised in this class with your fellow students. Don't even think about cheating.

**Grading:** 10% class participation, 40% Homework, 50% final project

**Calendar:** This calendar is fluid. I'll make changes based on our progress.

Note that January 29th is the last day to add this class and February 12th is the last day to drop this class.

Date	Topics	Examples	Homework
24-Jan	Anaconda Data types Variables Functions Running main()	HelloWorld	BuggyCode.py
31-Jan	local and global variables list for() if() else Files	none	Outlier remover
7-Feb	Structured Programming Using the Debugger Sets Dictionaries Data structures		Datacrunch
14-Feb	Randomization and simulation Gaussian Roll-your-own distributions		Retirement simulator Download and install Psychopy
21-Feb	Psychopy block randomization	Erikson Flanker task	Visual Search
28-Feb	Animation Objects	RSVP	FINST
7-Mar	Timing File directories and folders	PRP	Interruptions
14-Mar	<b>SPRING BREAK</b>		

21-Mar	Manipulating images Sound	TBA	TBA
28-Mar	Data analysis and graphing MAD Interfacing with R	TBA	TBA
4-Apr	Project discussion Gaze contingent eye tracking Interfacing with lab equipment	TBA	TBA
11-Apr	GUIs using <i>tkInter</i>	TBA	TBA
18-Apr	More stuff	TBA	TBA
25-Apr	Projects	Final Project Demonstrations	Final Project Demonstrations
2-May	Final Project Demonstrations	Final Project Demonstrations	Final Project Demonstrations
9-Mayy	Final Project Demonstrations	Final Project Demonstrations	Final Project Demonstrations