

# Python for Astronomy

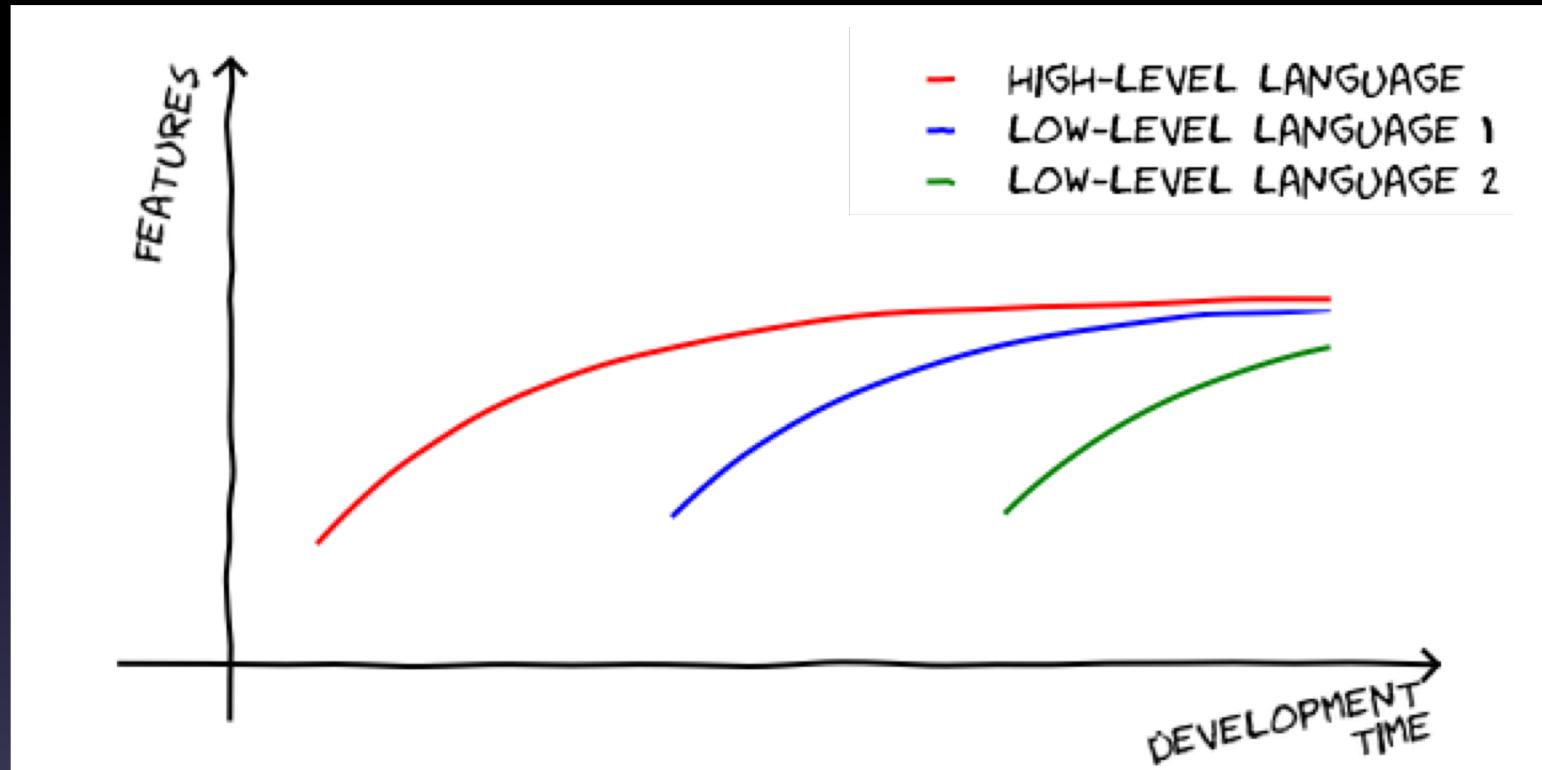
ASTR 302

[ajc@astro.washington.edu](mailto:ajc@astro.washington.edu)

# Why Python for Astronomers?

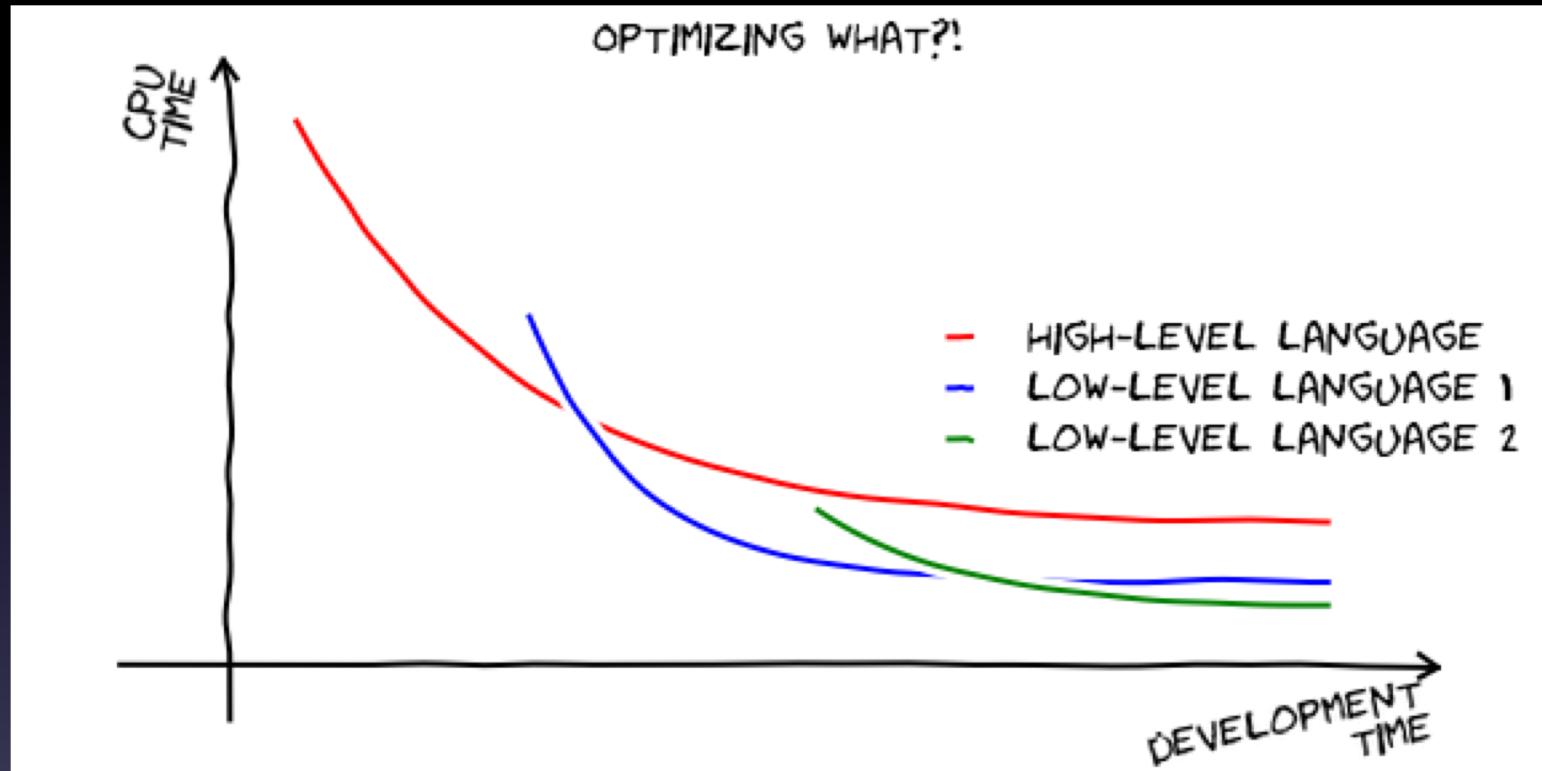
- *Python has recently become the common language of astronomy (specifically, astronomical data analysis).*
- The goal of this course is to teach how to effectively use the tools available in the Python ecosystem (the language, the libraries, and the concepts underpinning both) for your research. Think of it as a part of a series: ASTR 300 -> 302 -> 324 -> 497 (Astro Surveys & Big Data).
- While motivated by astronomy, the skills learned here should be broadly useful. It should easily transfer to more general data-science work.

# Why Python?



Higher-level languages, in general, shorten your time to research results. Python is one such language that is easy to learn, freely available, similar to legacy languages (IDL, FORTRAN, C/C++), and was mature enough at just the right time to spur adoption.

# As usual, lunches don't come free



Computational efficiency is the price you pay for speed of development. But that is typically a good trade to make.

# “The Only Thing That Is Constant Is Change”

- typically misattributed to Heraclitus

- In astronomical data analysis (and related computing disciplines), things change too quickly to be learned only once. Learning is a process that never ends.
- I will try to teach you what is currently the best (IMNSHO!) set of tools and techniques to have in your toolbox.
- But know this will change. So the major emphasis of this course will be on *understanding the concepts*, and learning how to continuously keep your knowledge up-to-date.



*Languages may change,  
but concepts transfer!*

# Learning how to Learn

- We'll also learn **how** to learn: how to discover information, keep your knowledge up-to-date, find leads that help you solve problems, evaluate their worthiness.
- This may be the most valuable piece to take away from this class.

# Lectures

- When: M-W, 11:00pm-12:30pm
- Where & how:
  - PAB 360
  - Can use the computers in the lab, or your own laptops.
    - If you're running Linux or have a Mac, I'd recommend the latter; it will let you (quite literally) take your work home with you!
- Lecture structure:
  - About an ~hour for introducing new material. **!!! WILL BE HANDS-ON !!!**
  - Leave ~20 minutes to work on homeworks, questions, open-ended discussion
  - Interrupt and ask questions at any time!

# Course Materials

- I'll be adding most of what we need to the following repository on GitHub:

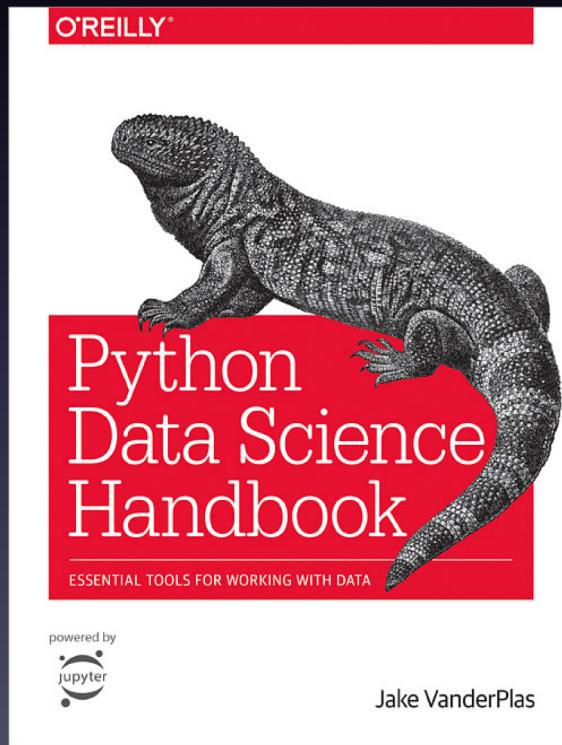
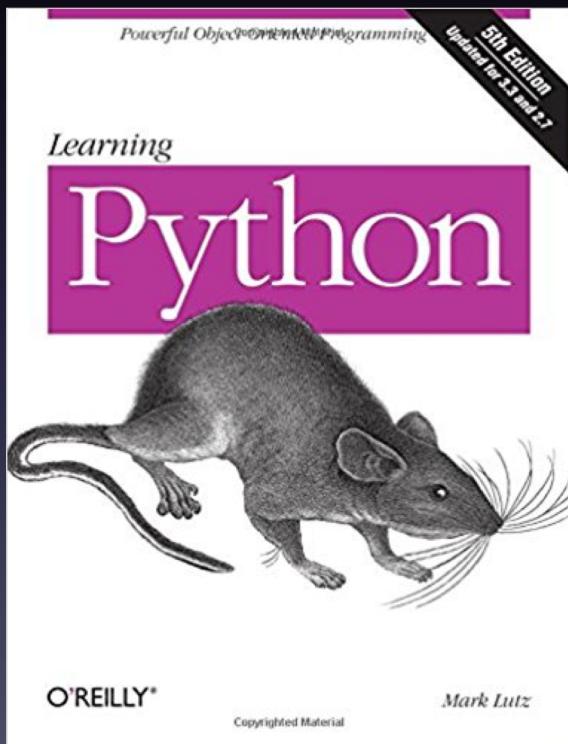


[https://github.com/connolly/A302\\_2019](https://github.com/connolly/A302_2019)

*Action: Let's make sure everyone can access this repository.*

# Textbooks & Reference Material

- There is a wealth of material available online and as eBooks (for UW students, via UW Libraries & Seattle Public Library). The two I'd recommend to begin with:



+ many (many) online writeups / blog posts that I will point you to over the next few weeks.

# Syllabus

- Available at  
[https://github.com/connolly/A302\\_2019/](https://github.com/connolly/A302_2019/)
  - (this will take you to the class github repository).
- Things to note:
  - The syllabus will be tweaked as we go through the quarter. Your feedback will be invaluable!

# Syllabus

When	Topic
Jan 7	Introduction and questionnaire
Jan 9	Introduction and questionnaire
Jan 14	Basic Python Refresher, Part I
Jan 16	Basic Python Refresher, Part II
Jan 23	How to be organized and collaborative: git and github
Jan 28	Interactive Data Analysis: Jupyter Project
Jan 30	Astronomical Python: Catalogs
Feb 4	Astronomical Python: Spectra and Images
Feb 6	Python Data Analysis Library: Pandas
Feb 11	Astronomical Python: Time Series
Feb 13	Database Introduction for Astronomers
Feb 18	Astronomical Data Archives
Feb 20	Using Databases and Archives from Python
Feb 25	Remote Data Analysis with Jupyter
Feb 27	Astronomical Python: Image Reduction
Mar 4	Astronomical Python: Object Detection and Measurement
Mar 6	When Your Code Starts to Grows: Basics of Software Engineering and Community Development
Mar 11	Astronomical Python: Machine Learning, Part I
Mar 13	Astronomical Python: Machine Learning, Part II
Mar 15	<i>Final Project Due</i>

# Grading, Homeworks, Etc.

## Seven homeworks (70% of the grade):

- Designed to exercise what we've learned recently.
- All homeworks will be turned in via GitHub

## Final project (30% of the grade):

- You will propose a piece of software (or analysis) to build using the techniques and libraries we'll learn about in the course. Keep an eye out for ideas!

*Advice: Turn in your homeworks on time!*

# Communication: Slack

- In this class, we won't be using a mailing list but an instant messaging (-like) tool called Slack (<http://slack.com>). Slack is heavily used today by many research & technology companies and projects.
- What to use it for:
  - Asking questions, discussing the class, exchanging snippets of code, collaborating on projects (when appropriate).
  - Feel free to create your own channels, but mostly use #general.
  - Please prefer Slack to sending me e-mails. Two reasons:
    - Everyone can benefit from the question and answer.
    - Your colleagues may be able to help!



*Action: Let's make sure everyone's on Slack.*

# Python: 2.7 vs. 3.X

- You've learned Python in ASTR 300. You may have heard about there being two major flavors (versions) of the language: Python 2 and Python 3
- From our point of view, the differences are (relatively) minor
  - See here for a nice overview:  
[http://sebastianraschka.com/Articles/2014\\_python\\_2\\_3\\_key\\_diff.html](http://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html)
- **In this class we'll use Python 3.** Python 2.7 is quickly falling out of use (and is not being actively developed). If at some point you need to use 2.7, the differences are sufficiently small that it will be easy to switch.

# Getting Python: Miniconda

- Let's install the Miniconda Python Distribution.

*<https://conda.io/miniconda.html>*

Note: in ASTR 300, you've used Jupyter (ipython) notebooks via a pre-installed Python distribution. Everything "just worked ™".

Here, I want to make sure you yourself know how to set up a Python environment from scratch: a) it's easy, and b) you won't always have sysadmins available to do it for you.



# Next week

## A Walk Through Python!