PROGRAMMING WITH PYTHON

GO.GWU.EDU/LIBWORKSHOPS

Instructors

- Dan Kerchner kerchner@gwu.edu
- Justin Littman justinlittman@gwu.edu

Materials: go.gwu.edu/pyw

Please sign in at: go.gwu.edu/laisignin

Today's Plan

~80 minutes:	~40 minutes
Basic Concepts	Data with Pandas

How to be awesome (for beginning programmers)

- Ask questions!
- If something is confusing in the workshop, it probably needs improvement; let us know.
- Google is your friend, especially as you continue on with Python

How to be awesome (for people with programming experience)

- Help each other out
- Respect every question and person asking the question

Adapted from slides by Terian Koscik @spine_cone

Ask us for help if you are lost/stuck/curious/confused!

Why Python?

- General purpose
- Easy to learn
- Readable*
- Open Source
- Widely Used
- Good built-in and contributed libraries

The Zen of Python

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.

read the rest at:
www.python.org/dev/peps/pep-0020

Different ways to use Python

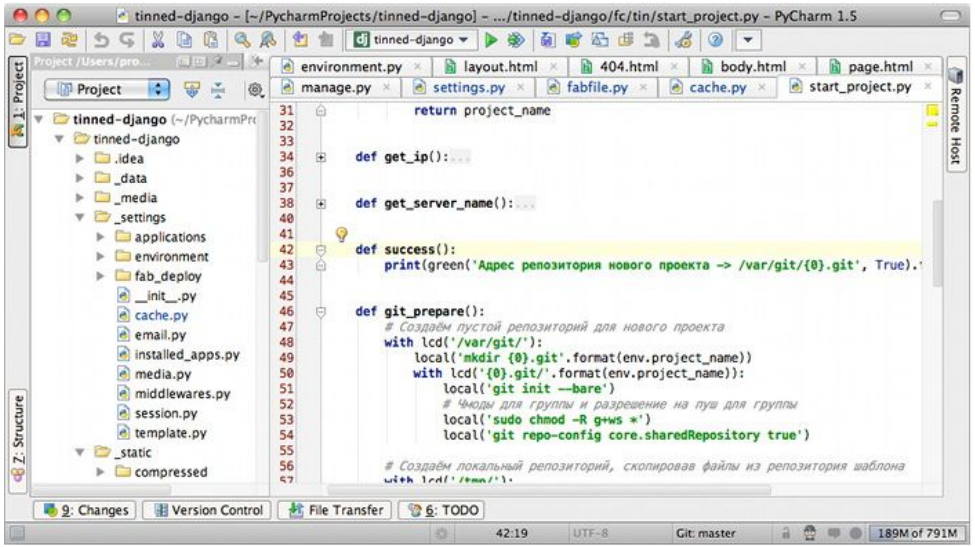
- Command line/REPL

```
Last login: Mon Mar 20 22:09:33 on ttys001
GLSS-M17LFFT:~ kerchner$ python
Python 2.7.10 (default, Oct 23 2015, 19:19:21)
[GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.0.59.5)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> opinion = "This workshop is awful!"
>>> opinion == True
False
>>>
```

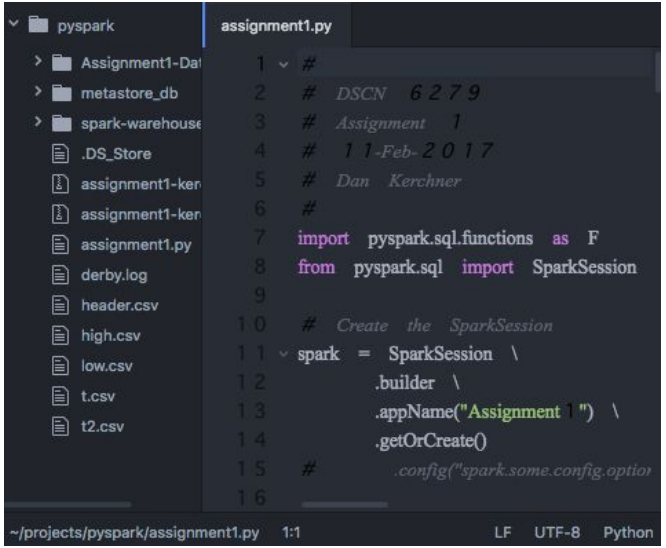
- Jupyter notebooks
  - Stay tuned...

Different ways to use Python

- Integrated Development Environment (IDE) – Spyder, pyCharm, pyDev, Sublime, …



- File editor (e.g. Atom, vim) + command line tools (pip, virtualenv, …)



Today's workshop

Anaconda = Python (and R) plus:

- **Jupyter notebooks**
- package management
- lots of libraries
  - data processing
  - analytics
  - scientific computing
  - including: **Pandas**

**Lessons based on Data Carpentry workshop*

Getting started

- Anaconda install
- Start up Anaconda Navigator, launch Jupyter notebook
- Download the files – instructions are at **go.gwu.edu/pyw** (a.k.a. github.com/gwu-libraries/gwlibraries-workshops/tree/master/python-programming)

Some recommendations

- Python 3
- Write assuming your code will be read (incl. by Future You)
- Version your code  GitHub
- Learn to be "Pythonic" in your style
- Use an editor with Syntax highlighting (most do)
- Isolate your environments
- Keep learning

Some Python libraries/frameworks

Building web applications	Django Flask
Scientific/numerical	Numpy Scipy Pandas
Machine Learning	scikit-learn
Data Visualization	matplotlib ggplot (like ggplot2 in R) plotly (<- interactive)

Tutorials & Help

- learnpython.org
- Software Carpentry, Data Carpentry (not just Python)
- Lynda.com lynda.it.gwu.edu courses: 13 Python, 3 Pandas
- More on Pandas:
  - http://pandas.pydata.org/pandas-docs/stable/10min.html
  - http://pandas.pydata.org/pandas-docs/stable/tutorials.html
  - http://pandas.pydata.org/pandas-docs/stable/cookbook.html
  - http://www.datacarpentry.org/python-ecology-lesson/
- Coding Consultations at GW Libraries – go.gwu.edu/coding

Contact us:

Dan Kerchner kerchner@gwu.edu
Justin Littman justinlittman@gwu.edu
Laura Wrubel lwrubel@gwu.edu

Workshop Materials: go.gwu.edu/pyw
Did you sign in? go.gwu.edu/laisignin