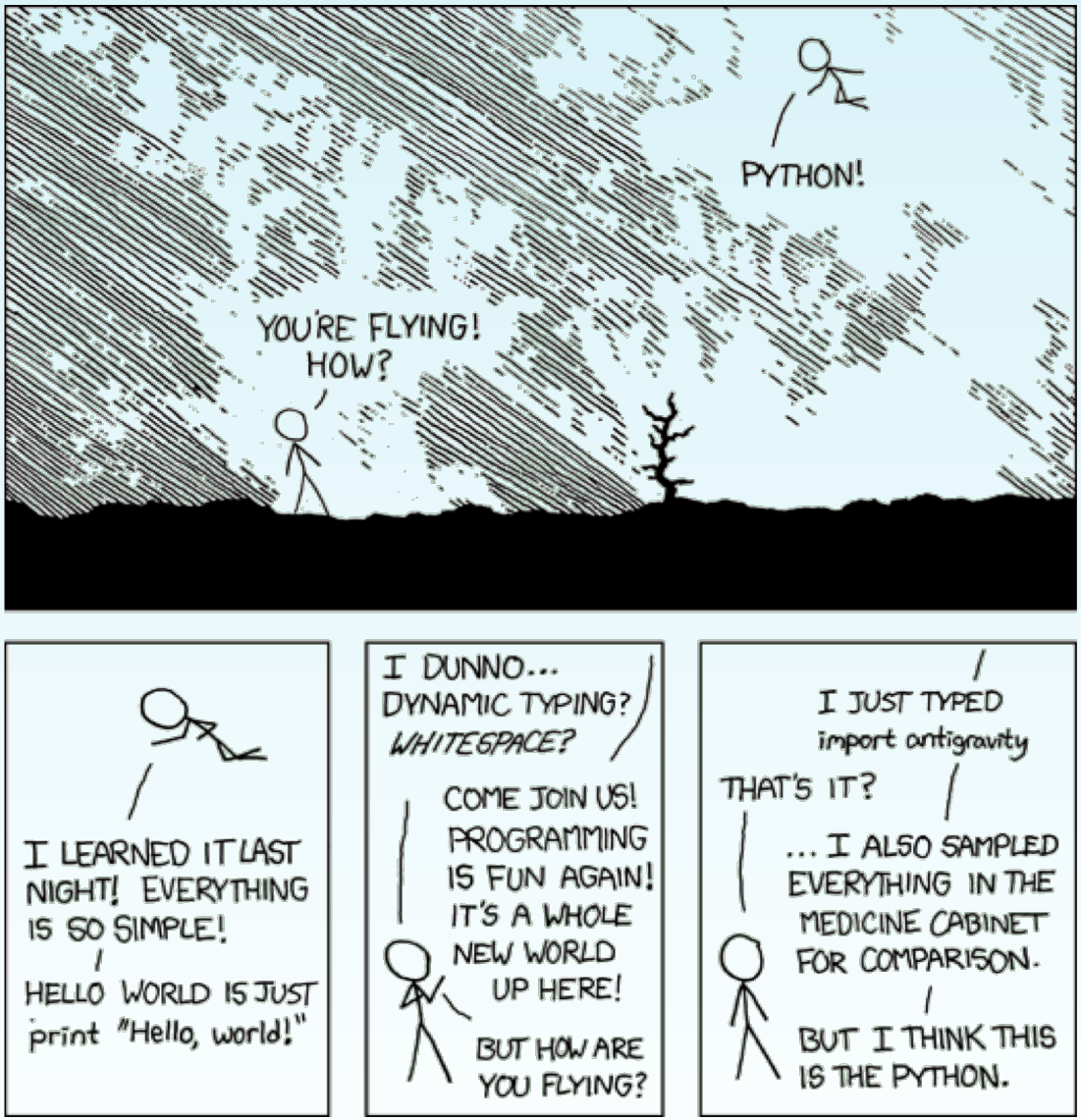


xkdc: A webcomic of romance, sarcasm, math, and language.

Object-Oriented Computer Programming for GIS

Lecture 0: Syllabus/Introduction

Dr. Giovanni Baiocchi[1]

September 2014, College Park

[1] Department of Geographical Sciences, UMD.

Outline

Course Instructor, Teaching Assistant

- Instructor
  - Dr. Giovanni Baiocchi, Associate Professor at the Geography Department UMD
    - Email: baiocchi@umd.edu
    - Office hours: Fridays 2–3 pm, room 1133
- Teaching assistant
  - Patrick McDonough
    - Email: p.mcdonough8133@gmail.com
    - Office hours: Wednesday 12:30-2:00pm, Thursday 1:30–3:00pm, Teaching Assistant room

Course Aims

DEVELOP GENERAL COMPETENCY IN:

- Writing simple code using an Object-Oriented programming language
- Automating components of work in IPython
- Conducting customized analyses of geospatial data sources

Course objectives

- Develop a deeper understanding of programming techniques (applied to Python)
- Expand programming capabilities within GIS
- Develop proficiency in using a modern computational environment to automate processing of geospatial data
- Develop ability to problem-solve independently
- Introduce more advanced Python and geospatial Python-based open-source packages used to process geospatial information and rapidly expanding new GIS open-source data formats
- Acquire necessary information from help files

Course scope

- Object-oriented programming (Python and GIS packages)
- Developing custom applications for geospatial data manipulation and analysis
- Prerequisites: GEOG 373 or another GIS course, GEOG 376 (former GEOG398A) or another basic programming course

Main Software

- The required software for this class is **Python**. Python is the open source and freeware, one of the most powerful and versatile programming languages, and is available for free download for use on PC, Mac, UNIX and Linux environments.
- The recommended installation is **Anaconda**, available at **here**
- For Python I will use the **IPython notebook**, a web-based interactive and computational environment **here**
- If you install Anaconda's Python distribution, available for most platforms, the notebook and most **useful packages** will be already installed, including the IPython Notebook (NumPy, SciPy, Pandas, IPython, Matplotlib).
- If you are already using **other Python** distribution I do not recommend installing Anaconda on top of it (unless are a consummate programmer!)
- Other **open-source GIS packages** needed can be installed on demand when required depending on the distribution and operating system you use.
- The software is available in the Lefrak/Geography **Open Lab** on the PC machines. If you have a laptop or home computer, get this software and download it immediately. Object-oriented programming (Python and GIS packages)

Main Python Packages (already installed)

- **The IPython Notebook**. The IPython Notebook is a web-based interactive computational environment where you can combine code execution, text, mathematics, plots and rich media into a single document
- **NumPy**. NumPy is the fundamental package for scientific computing with Python
- **Python Data Analysis Library**. pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language
- **matplotlib**. matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms

GIS-based Python Packages: might need separate installation

- **pyproj**. Performs cartographic transformations and geodetic computations.
- **fiona**. Fiona provides uncomplicated Python interfaces to functions in OGR, the best open source C/C++ library for reading and writing geographic vector data.
- **GDAL**. GDAL: Geospatial Data Abstraction Library
- **geopy**. geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources
- **Shapely**. Shapely is a BSD-licensed Python package for manipulation and analysis of planar geometric objects. It is based on the widely deployed GEOS (the engine of PostGIS) and JTS (from which GEOS is ported) libraries.
- **PyShp**. This library reads and writes ESRI Shapefiles in pure Python. You can read and write shp, shx, and dbf files with all types of geometry. Everything in the public ESRI shapefile specification is implemented. This library is compatible with Python versions 2.4 to 3.x.
- **Basemap**. Module for plotting data on maps with matplotlib
- **Descartes**. Use Shapely or GeoJSON-like geometric objects as matplotlib paths and patches
- **PySAL**. PySAL is an open source library of spatial analysis functions written in Python intended to support the development of high level applications.
- **GeoPandas**. GeoPandas is an open source project to make working with geospatial data in python easier. GeoPandas extends the datatypes used by pandas to allow spatial operations on geometric types. Geometric operations are performed by shapely. Geopandas further depends on fiona for file access and descartes and matplotlib for plotting.
- **Rtree**. Rtree is a ctypes Python wrapper of libspatialindex that provides a number of advanced spatial indexing features for the spatially curious Python user
- **folium**. Folium builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the Leaflet.js library
- **Mapnik**. Mapnik is a Free Toolkit for developing mapping applications. Above all Mapnik is about making beautiful maps.
- **Vincent**. The data capabilities of Python. The visualization capabilities of JavaScript.

Attendance policy, assignments, etc.

- Attendance is required for lectures, labs, mid-term, project presentations, and the final exams
- Lab assignments are given on the day Labs are held and are due the next lab session, all due date modifications will be announced and posted on Blackboard, submission of assignments through course folder
- Lab assignments and collaboration :
  - These are assignments to be completed individually!!!!
  - Program design and implementation MUST be completed independently
  - Minor help with debugging is OK.

Lecture structure

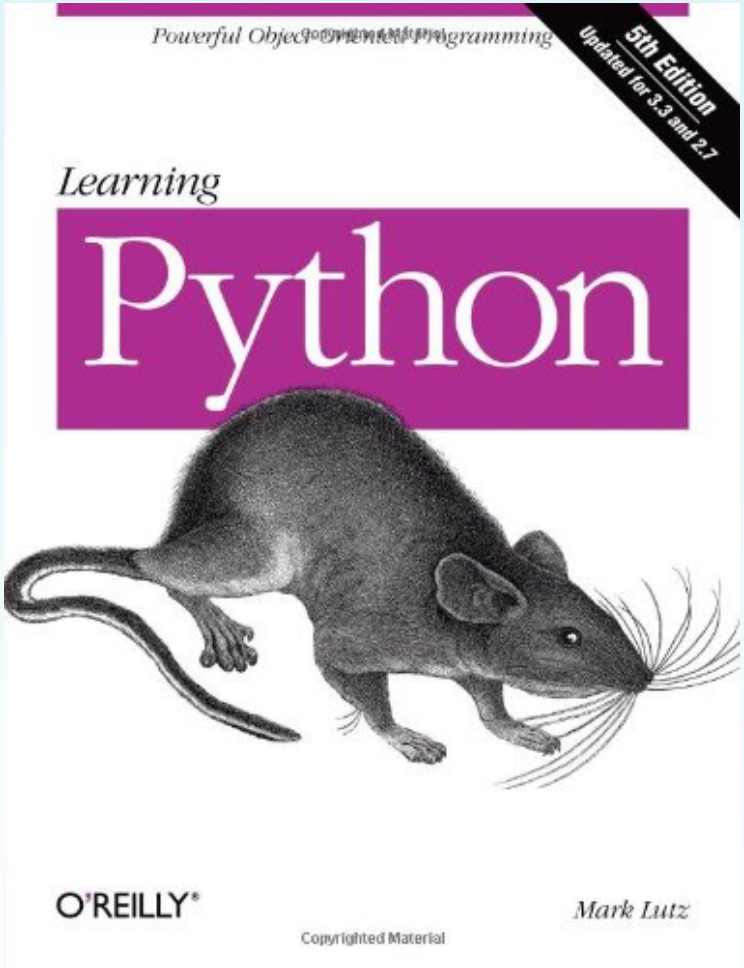
- One two=hours lecture per week (Tue 12:00 pm – 14:00 pm, room LEF 1124)
- 2 sections:
  - Concepts (lecture-like)
  - Implementation + exercises (hands-on) - you are encouraged to bring your laptop and practice writing code in the classroom
- I will NOT post lecture notes - be prepared to take very good notes during the class

Lab structure

- One Lab per week
  - Wednesday or Thursday 3 pm-5 pm, room LEF 1138
- Lab preparation:
  - Labs are focused on using Python for problem solving and are likely to take longer than 2 hours to complete if the subject is not examined before the lab
  - If applicable, come to the lab with most of your program pre-written in pseudo code that will speed things up
  - TA will address important issues and concepts at the beginning of each lab, come on time - he will not give a special encore presentation just for you

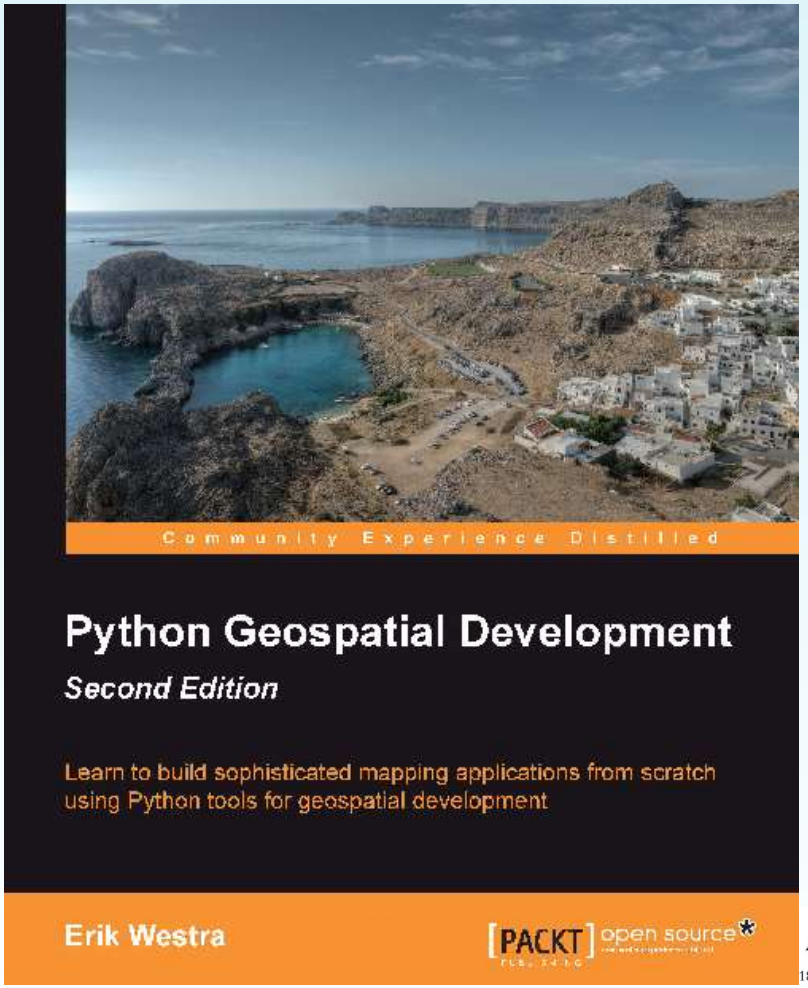
Recommended/Reference Textbooks

- Mark Lutz, Learning Python, 5th Edition
  - Get a comprehensive, in-depth introduction to the core Python language.
  - This self-paced tutorial gets you started with both Python 2.7 and 3.3- lines.



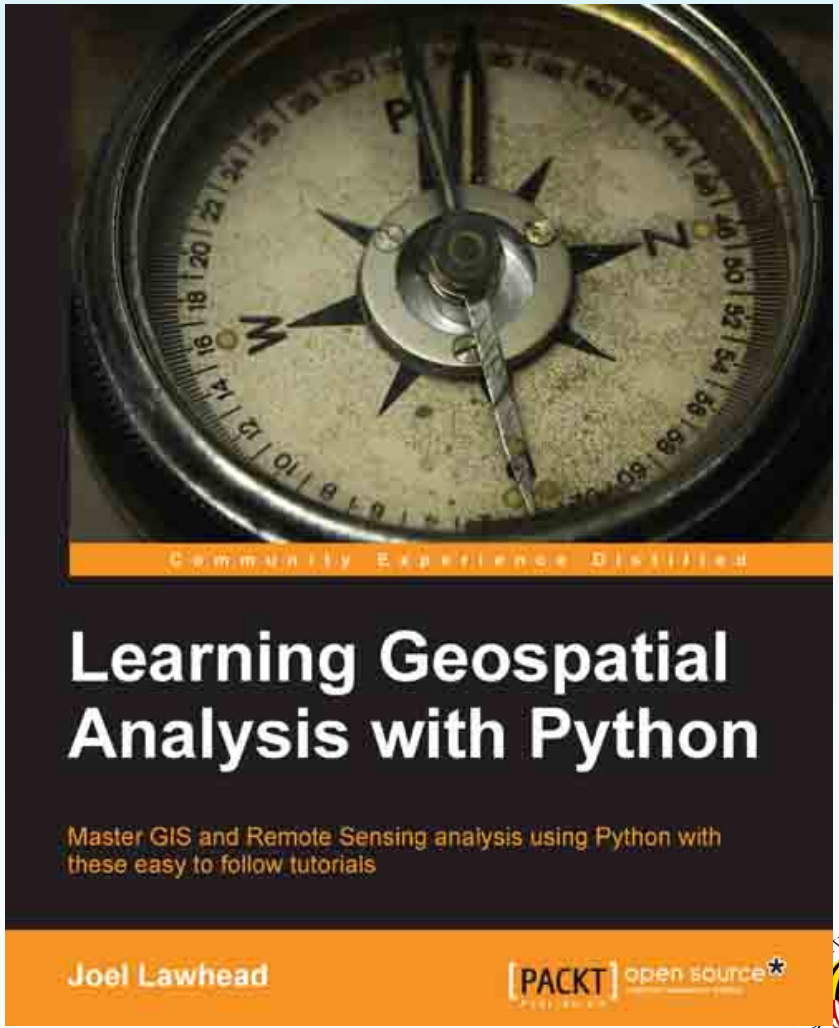
Recommended/Reference Textbooks

- Mark Lutz, Learning Python, 5th Edition
  - Get a comprehensive, in-depth introduction to the core Python language.
  - This self-paced tutorial gets you started with both Python 2.7 and 3.3- lines.
- Erik Westra, Python Geospatial Development - Second Edition
  - Build your own complete and sophisticated mapping applications in Python.
  - Walks you through the process of building your own online system for viewing and editing geospatial data
  - Practical, hands-on tutorial that teaches you all about geospatial development in Python



Recommended/Reference Textbooks

- Mark Lutz, Learning Python, 5th Edition
  - Get a comprehensive, in-depth introduction to the core Python language.
  - This self-paced tutorial gets you started with both Python 2.7 and 3.3- lines.
- Erik Westra, Python Geospatial Development - Second Edition
  - Build your own complete and sophisticated mapping applications in Python.
  - Walks you through the process of building your own online system for viewing and editing geospatial data
  - Practical, hands-on tutorial that teaches you all about geospatial development in Python
- Joel Lawhead, Learning Geospatial Analysis with Python
  - Construct applications for GIS development by exploiting Python
  - Focuses on built-in Python modules and libraries compatible with the Python Packaging Index distribution system - no



Outline

Recommended/Reference Textbooks

Introduction, Basic Tools				
	Key Date		**Event**	**Topic**
Week 1	Tue 02-Sep	12–14	**Lecture 1**	Introduction
	Wed 03-Sep	15–17	LAB: 1	IPython Notebook
Week 2	Tue 09-Sep	12–14	**Lecture 2**	Review Basic Python
	Wed 10-Sep	15–17	LAB: 2	Basic Python
Advanced Python for GIS				
Week 3	Tue 16-Sep	12–14	**Lecture 3**	Lists, Tuples, Dictionaries, and Sets
	Wed 17-Sep	15–17	LAB: 3	Practice: Advanced Python Data Structures
Week 4	Tue 23-Sep	12–14	**Lecture 4**	Control flows, Iterators
	Wed 24-Sep	15–17	LAB: 4	Practice: Control flows, Iterators
Week 5	Tue 30-Sep	12–14	**Lecture 5**	Subroutines, λ-functions, Generators, Creating libraries
	Wed 01-Oct	15–17	LAB: 5	Subroutines, λ-functions, Generators, Creating libraries: Convert Geographical Coordinates

Recommended/Reference Textbooks

Core GIS with Python				
Week 6	Tue 07-Oct	12–14	**Lecture 6**	Pandas for Data Analysis (Pandas)
	Wed 08-Oct	15–17	LAB: 6	Practice: Pandas, Lab5 assigned
Week 7	Tue 14-Oct	12–14	MIDTERM I (in class: 1 hour, Starts at 12)	
	Wed 15-Oct	15–17	No Lab	
Week 8	Tue 21-Oct	12–14	**Lecture 7**	Geodetic Computations: pyproj, GeoPy, GeographicLib. Reading and Processing Shape files, Map Projections: Fiona, GDAL/OGR
	Wed 22-Oct	15–17	LAB: 7	Practice: pyproj, GeoPy, GeographicLib, Fiona, GDAL/OGR
Map Visualization with Python				
Week 9	Tue 28-Oct	12–14	**Lecture 8**	Python 2D plotting library: Matplotlib
	Wed 29-Oct	15–17	LAB: 8	Practice: Matplotlib
Week 10	Tue 04-Nov	12–14	**Lecture 9**	Maps in Python: Basemap, Descartes
	Wed 05-Nov	15–17	LAB: 9	Practice: Basemap, Descartes
Week 11	Tue 11-Nov	12–14	**Lecture 10**	Rendering Maps with Mapnik
	Wed 12-Nov	15–17	LAB: 10	Practice: Mapnik

Recommended/Reference Textbooks

Processing Geospatial Data with Python				
Week 12	Tue 18-Nov	12–14	**Lecture 11**	Shapely
	Wed 19-Nov	15–17	LAB: 11	Practice: Shapely
Week 13	Tue 25-Nov	12–14	**Lecture 12**	Review/Catch up
	Wed 26-Nov	15–17	LAB: 12	Practice: Shapely, Pandas, PySAL
Week 14	Tue 02-Dec	12–14	**Lecture 13**	Processing Geospatial Data with Python
	Wed 03-Dec	15–17	LAB: 13	Course Project Due
Week 15	Tue 09-Dec	12–14	Review	
	Wed 10-Dec	15–17	FINAL EXAM (in LAB)	

Outline

Independent project

- Highlight your abilities and accomplishments in this course
- Come up with and implement a practical or fun program written by you in Python and focused on geospatial data analysis
  - This might end up being a group project but each person in the group should have a specific piece of code she/he is

Exams

- Midterm exam (held in the lab):
  - Most likely a set of python code implementations for a subset of smaller tasks demonstrating the proficiency in object oriented programming
- Final exam (held in the lab):
  - Most likely an overall exercise in program design and implementation with a specific focus on geospatial data analysis.

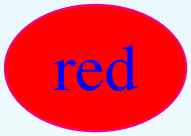
Grading

- Lab assignments (50%)
- Mid-term exam (15%)
- Independent project (15%)
- Final exam (20%)

Outline

Questions

- Questions
- Other issues
- You take the 

or 

pill?