GIS 4653/5653: Spatial Programming and GIS

Valliappa Lakshmanan

lakshman@ou.edu

NWC 4457

What is this course about?

- Being able to write programs for spatial analysis
  - Why do you have to write programs? Why not just use ArcGIS?

Why Learn Spatial Programming?

- Interactive programs have serious limitations
  - Need humans involved at each stage
  - Costly, slow, greater chance of error
- Automated programs can help in some of these situations:
  - Routine (repetitive) tasks
  - Analyze streaming data
  - Decision support (what-if scenarios)
  - Precomputation

The Personal Computer

> *I think it's fair to say that personal computers have become the most empowering tool we've ever created. They're tools of communication, they're tools of creativity, and they can be shaped by their user.*
>
> Bill Gates, founder of Microsoft

Image © San Jose Library

Are Washing Machines Computers?

- Modern washing machines have microprocessor controls
  - Control the times, water flow, etc.

- Have you used a washing machine
  - For communication?
  - To create something?
- Have you “shaped” a washing machine?

Using a Computer

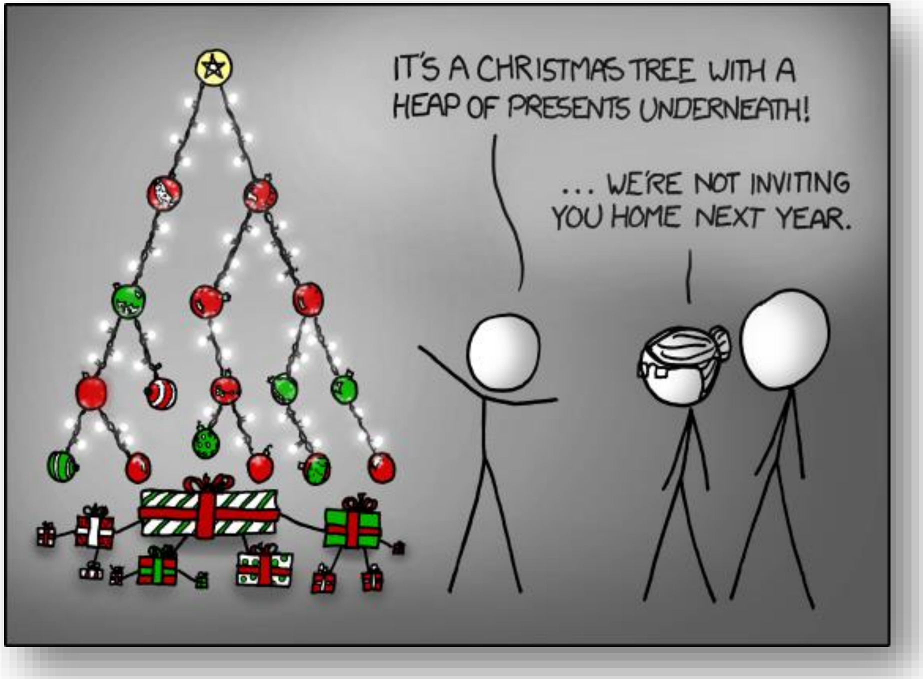
- How about a personal computer? Have you:
  - Used it for communication?
  - Used it to create something? If so, what?
  - Did you shape it?
- How do you "shape" a computer?

A Programmable Instrument

- Astonishing variety of things you can do with a computer
  - Because computers are programmable
  - Some one created the applications you use
- To "shape" a computer is tell it how to do what you want
  - To play games, to process documents, to display presentations, etc.
- Can you program a washing machine?
  - How is programming a personal computer different?

Programmers

- Millions of computer users have never programmed a computer
  - They use programs developed for their machines
- Programmers tell the computer how to do the things that users want


IT'S A CHRISTMAS TREE WITH A HEAP OF PRESENTS UNDERNEATH!
... WE'RE NOT INVITING YOU HOME NEXT YEAR.

http://xkcd.com/835/

What this course is about

- This course is about taking you beyond interactive GIS
  - We will do spatial programming
  - Gain the ability to tell a computer to do a GIS task in the context of a larger program

Logistics

Hours

- 3 credit hours per week = 120 min
  - Class meets Fridays from 1 pm to 3.40pm (w/breaks)
  - SEC N0108
- Office hours at NWC 4457 on Thursdays 1pm to 4pm
  - Feel free to drop in to my office on other afternoons
  - Or make an appointment via email

Textbook and slides

- No textbook
  - Should become familiar with reading online documentation of the various software packages we will use
  - The majority of your learning will happen through reading, programming and experimentation. Class work and homework are meant to get you thinking and are an integral part of this course.

- All my slides will be placed online by the time the class starts:
  - http://cimms.ou.edu/~lakshman/

My References

- Python
  - Learning Python by Mark Lutz, O'Reilly Press
  - Programming Python by Mark Lutz, O'Reilly Press
- Open-source GIS in Python
  - Python Geospatial Development by Erik Westra, Packt Publishing
  - http://www.gis.usu.edu/~chrisg/python/2009/lectures
- Python in ArcGIS
  - Python Scripting for ArcGIS by Paul Zandbergen, ESRI Press
  - https://www.e-education.psu.edu/geog485/

- I will try to strike a balance between ESRI and open-source
  - My sympathies learn towards O-S, so you will find this leaking out
  - I do know, however, that the ESRI product is a lot more convenient

Grading policy

- Undergraduate students: course grade will be based on homework (70% of grade) and an exam (30%).
- Graduate students: course grade will be based on homework (35%), a term project (35%) and an exam (30%).
- Letter grades are based on absolute points and will not be curved: A (>= 90%), B (>= 80%), C (>= 70%), D (>= 60%), F (< 60%).

- Each homework and term project is graded on a scale of 10 points. Late submissions are subject to a penalty of 2 points off per day. No credits will be given for assignments or projects that are late for more than 5 days, unless granted prior permissions.

Exams and Reports

- The exam will cover concepts covered in the class; the exact format of the exam (multiple-choice/short-answer/computer-project etc.) is subject to change.
  - It will be December 6 during class hours (NOT finals week)
- The topic of the term project (graduate students only) will be chosen by the student and approved by the instructor; the term project should be decided upon sometime after the mid-point of the semester.
  - You will have to present your work as a 12-minute talk

- All work should be submitted digitally (learn.ou.edu).
  - Reports should be in PDF (Microsoft Word, Powerpoint, etc. are not acceptable)
  - Always attach relevant code with comments
  - *Organize assignments so that you have intermediate outputs.* ***You will not receive any credit for partial code****, only for partial outputs.*

Integrity

- Each student in this course is expected to abide by Academic Integrity

- You are encouraged to study together and to discuss information and concepts covered in lecture and the sections with other students. You can give "consulting" help to or receive "consulting" help from such students on lab assignments or activities.

- However, this permissible cooperation should never involve one student having possession of a copy of all or part of work done by someone else.

Accommodations

- Any student in this course who has a disability that may prevent him or her from fully demonstrating his or her abilities should contact the instructor personally as soon as possible so accommodations can be made to ensure full participation and facilitate your educational opportunities.

- It is the policy of the University to excuse absences of students that result from religious observations and to provide without penalty for the rescheduling of examinations and additional required class work that may fall on religious holidays.

Introductions

Valliappa Lakshmanan (Lak)

- Email is the best way to contact me: lakshman@ou.edu
- Research Scientist at CIMMS since 1995
  - Research focuses on statistical/machine intelligent algorithms for severe weather detection, diagnosis and prediction
  - All my work is geospatial
- First course I am teaching in GES
  - Taught a cross-listed Meteorology/GIS course on Automated Analysis of Spatial Grids in 2011, 2013
  - Can download PDF of book based on that course for free at http://www.springer.com/engineering/signals/book/978-94-007-4074-7 as long as you are on university network

You …

- Your name
  - And how you prefer to be called if not by your full first name
- Something about what you are studying
  - Department? Undergraduate? Graduate? Have a thesis topic?
- What got you interested in spatial programming?
  - Is there something you hope to be able to achieve using spatial programming?

Programming Environment

Automation and ArcGIS

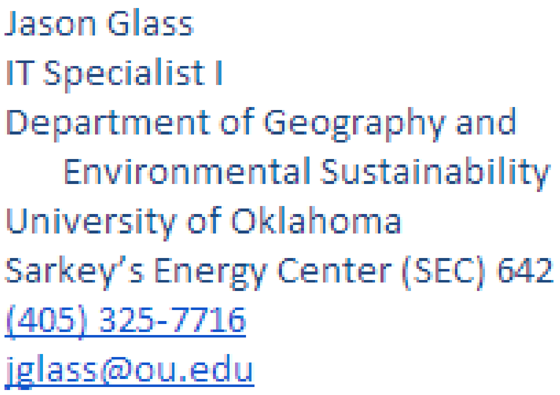
- Two ways of automating programs with ArcGIS
  - Model Builder for simpler tasks
  - Python tools for flexibility
  - We will do both
- But step back and consider whether you need ArcGIS
  - Many simple GIS tasks can be accomplished without ArcGIS
  - Open source GIS modules are significantly more flexible
    - Run on Linux
    - Can run as web services or within web servers on demand
    - Scale up to large numbers of users
    - Easier to "mash up" with other sources of GIS information
    - Cheap/free, no licensing issues

Python

- You will learn how to program in Python
  - Python is a programming language
  - Easy to learn
- Can "drive" ArcGIS using Python
  - Many of the ArcGIS toolkits are written using Python
  - Can also do many simple analysis and display tasks directly using Python and Python modules
    - No need for ArcGIS
- Python is slower than "compiled" languages
  - May be inefficient for big or streaming data

Software You Need

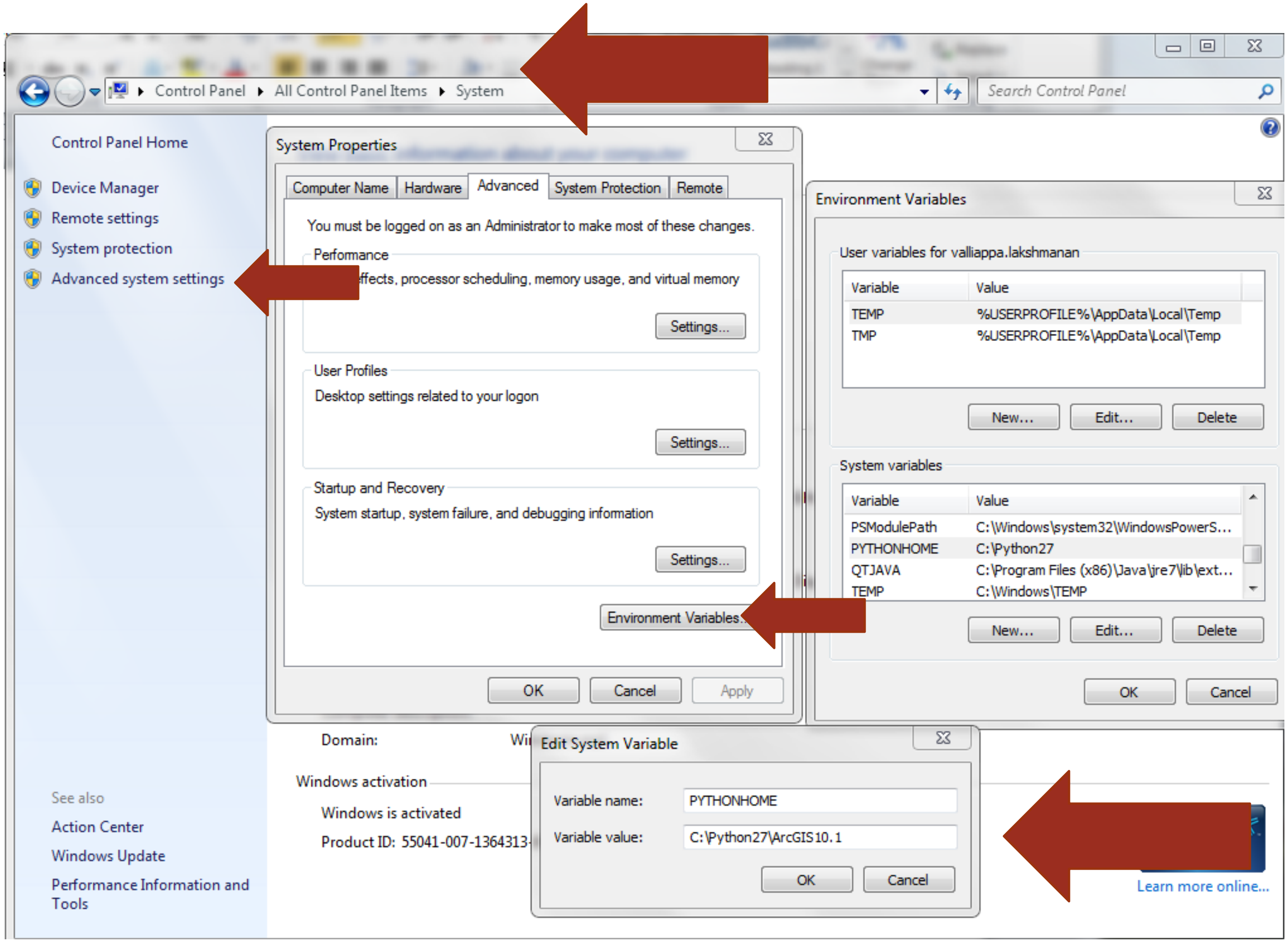
- ArcGIS Desktop 10.1 + Python 2.7
  - Get Installation CD from



Jason Glass
IT Specialist I
Department of Geography and
Environmental Sustainability
University of Oklahoma
Sarkey's Energy Center (SEC) 642
(405) 325-7716
jglass@ou.edu

  - Install the software on your laptop

PYTHONHOME



You must be logged on as an Administrator to make most of these changes.

Performance

…ffects, processor scheduling, memory usage, and virtual memory

User Profiles

Desktop settings related to your logon

Startup and Recovery

System startup, system failure, and debugging information

Domain:

Windows activation

Windows is activated

Product ID: 55041-007-1364313-

User variables for valliappa.lakshmanan

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

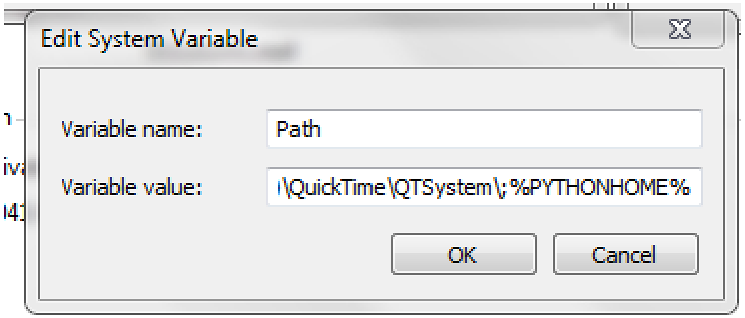
Variable	Value
PSModulePath	C:\Windows\system32\WindowsPowerS...
PYTHONHOME	C:\Python27
QTJAVA	C:\Program Files (x86)\Java\jre7\lib\ext...
TEMP	C:\Windows\TEMP

Variable name: PYTHONHOME

Variable value: C:\Python27\ArcGIS10.1

Path

- To be able to run Python from command-line

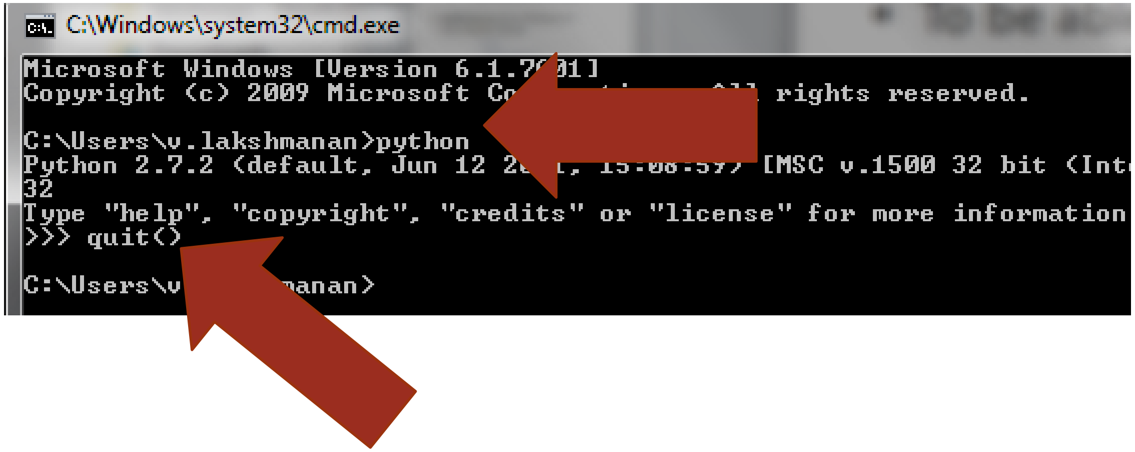


Edit System Variable

Variable name: Path

Variable value: \QuickTime\QTSystem\;%PYTHONHOME%

- Open up a command-prompt and type:



C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Co                All rights reserved.

C:\Users\v.lakshmanan>python
Python 2.7.2 (default, Jun 12 20                 [MSC v.1500 32 bit (Int
32
Type "help", "copyright", "credits" or "license" for more information
>>> quit()

C:\Users\v          manan>
```

Matplotlib

- Download 32-bit version for Windows:
  - https://downloads.sourceforge.net/project/matplotlib/matplotlib/matplotlib-1.2.1/matplotlib-1.2.1.win32-py2.7.exe

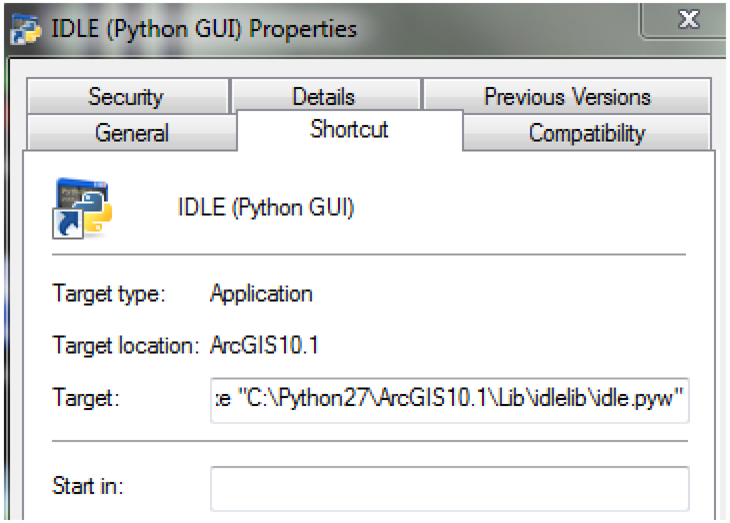
- Some newer versions of ArcGIS install both numpy and Matplotlib, so you may be able to omit this
  - The installation program of ArcGIS tells you which Python modules it is installing

Basemap

- Install latest 32-bit stable version using Windows installer:
  - http://sourceforge.net/projects/matplotlib/files/matplotlib-toolkits/basemap-1.0.6/basemap-1.0.6.win32-py2.7.exe/download

Python Programming Shell

- Soft-link the IDLE Python GUI that comes with Python



IDLE (Python GUI) Properties

IDLE (Python GUI)

Target type: Application

Target location: ArcGIS10.1

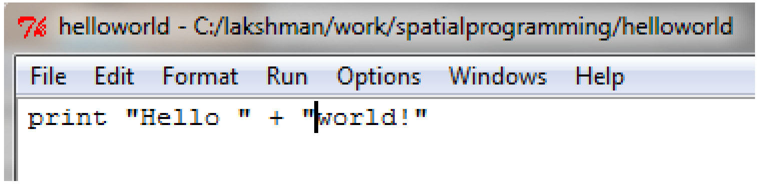
Target: :e "C:\Python27\ArcGIS10.1\Lib\idlelib\idle.pyw"

Start in:

- Start the program
- If you are familiar with some other shell, you can use that instead (use of IDLE is not mandatory)

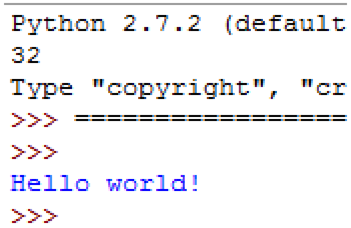
Your first program

- In IDLE, File | New Window
- In the window, type:



```
print "Hello " + "world!"
```

- Save the file somewhere and hit F5 (Run)

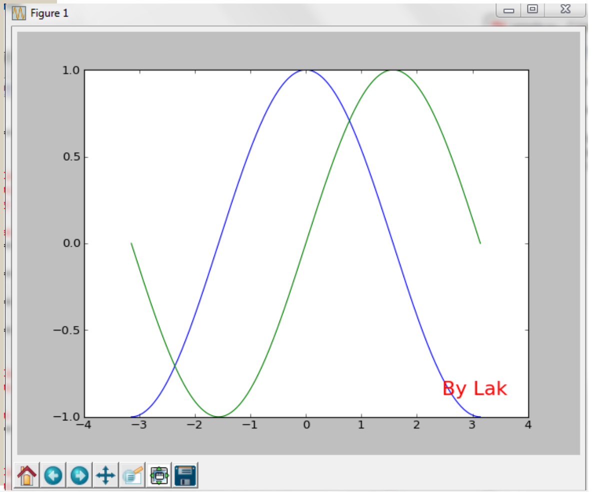


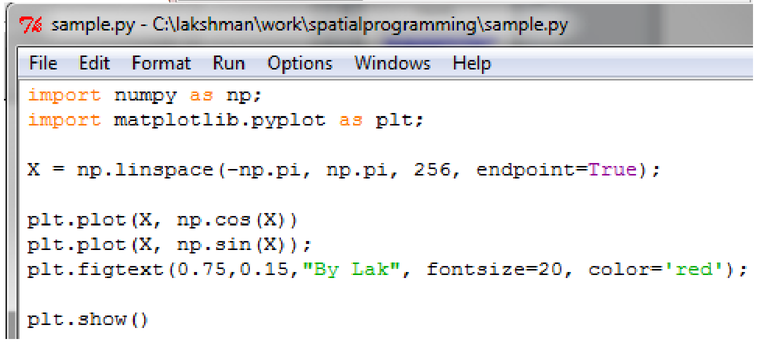
```
Python 2.7.2 (default
32
Type "copyright", "cr
>>> =================
>>>
Hello world!
>>>
```

Plotting

- See if matplotlib is installed properly
  - (Copy and paste from PDF to avoid typos)

```
import numpy as np;
import matplotlib.pyplot as plt;
X = np.linspace(-np.pi, np.pi, 256, endpoint=True);
plt.plot(X, np.cos(X))
plt.plot(X, np.sin(X));
plt.figtext(0.75,0.15,"By Lak", fontsize=20, color='red');
plt.show()
```



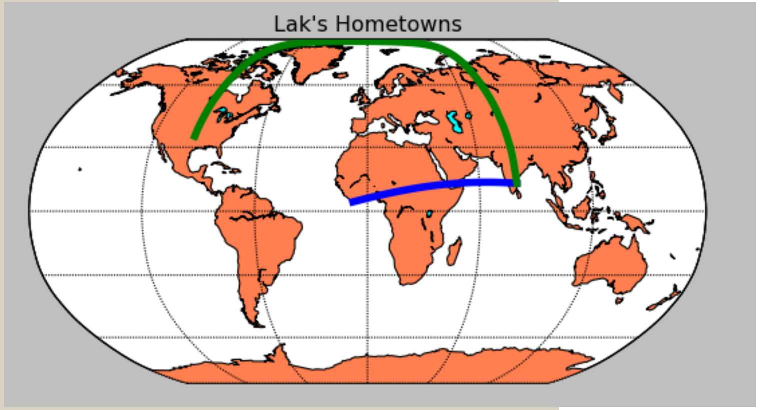


```
import numpy as np;
import matplotlib.pyplot as plt;

X = np.linspace(-np.pi, np.pi, 256, endpoint=True);

plt.plot(X, np.cos(X))
plt.plot(X, np.sin(X));
plt.figtext(0.75,0.15,"By Lak", fontsize=20, color='red');

plt.show()
```

Show Spatial Data

- Check that basemap is installed properly

```
from mpl_toolkits.basemap import Basemap
import numpy as np
import matplotlib.pyplot as plt
# lon_0 is central longitude of projection.
# resolution = 'c' means use crude resolution coastlines.
m = Basemap(projection='robin',lon_0=0,resolution='c')
m.drawcoastlines()
m.fillcontinents(color='coral',lake_color='aqua')
# draw parallels and meridians.
m.drawparallels(np.arange(-90.,120.,30.))
m.drawmeridians(np.arange(0.,360.,60.))
m.drawmapboundary(fill_color=‘white')
# draw a couple of great circles
m.drawgreatcircle(-7.72, 4.38, 80, 13.08, linewidth=5 );
m.drawgreatcircle(80, 13.08, -97.44, 35.22, linewidth=5 );
plt.title("Lak's Hometowns")
plt.show()
```

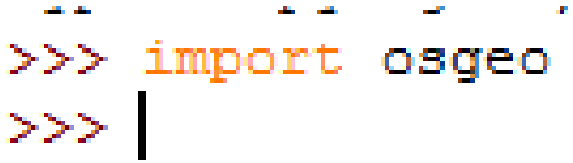


GDAL/OGR

- The final 3rd party Python package that we will use is GDAL/OGR
- Download and install the latest release version of the core package and the Python 2.7 bindings from http://www.gisinternals.com/sdk/
  - I installed GDAL v 1.10 core from http://www.gisinternals.com/sdk/Download.aspx?file=release-1600-gdal-1-10-0-mapserver-6-2-1\gdal-110-1600-core.msi and the Python binding http://www.gisinternals.com/sdk/Download.aspx?file=release-1600-gdal-1-10-0-mapserver-6-2-1\GDAL-1.10.0.win32-py2.7.msi

GDAL set up

- Install the core API using the installer
- Then, add the GDAL library to your PATH
  - I added: C:\Program Files (x86)\GDAL (remember to add a semicolon after the previous entry to separate it from this one)
  - To modify the path, change the Environment Variable Path
- Then, run the Python installer
- Open up a Python IDLE prompt and make sure this works:



```
>>> import osgeo
>>> 
```

Homework

- Install all necessary software on your laptop (or find GIS lab)
- Turn in as a PDF document a report containing:
  - A pair of curves plotted using matplotlib
    - Do some other curve other than sin, cos (perhaps sin+cos or exp?)
  - A plot of a journey or hometowns using Basemap
    - Look up the available projections in Basemap and use an appropriate one
    - I used the Robinson projection
  - The output of typing osgeo.version_info at Python prompt
  - Submit to learn.ou.edu
  - Customize the output to "prove" it is yours
- Due before start of next class