

New Course for Fall 2009

GIS Programming Practicum



Using Python scripting with ArcGIS

Python scripting is a powerful tool for automating many geoprocessing tasks. This course will take a hands-on approach to creating your own scripts.

Geo 599

2 credits, Monday 4:00-5:20

CRN: 18970

Topics will include:

- Exporting Model Builder models as scripts
- Running ArcGIS geoprocessing tools in scripts
- Conditional and Iterative statements
- Basic Python data types: Lists and dictionaries
- Object-oriented programming concepts

Instructor: Tracy Kugler

Students should be familiar with ArcGIS and geodatabases.

No scripting/programming experience is necessary, but students should be comfortable with experimenting.

GEO 599, Special Topics:
GIS Programming Practicum
Fall 2009

Course Information

2 Credit hours, CRN: 18970

Course meetings: Mondays, 4:00 – 5:20, Wilkinson 203

Prerequisites: GEO 465/565, or equivalent experience using ArcGIS

This course is intended to provide a practical introduction to the using the Python scripting language to automate geoprocessing in ArcGIS. Scripts are a convenient way to automate repetitive tasks in ArcGIS, allowing the computer to do the tedious work for you and maintain consistency. Python is a general-purpose scripting language, which ESRI has adopted as the standard for ArcGIS scripting.

The course will include an introduction to basic programming concepts, and key features of Python. Naturally, we will focus on applying these concepts to geoprocessing tasks, including running ArcGIS tools using scripts, working with attribute tables, creating and editing geometries, reading and writing text files, and working with rasters.

Instructor Information

Tracy Kugler, Geography program PhD Candidate, Campus GIS TA

kuglert@geo.oregonstate.edu

Office hours: Tuesday 9:00-10:00 and Thursday 6:00-7:00

in Digital Earth (Wilkinson 210)

Individual meetings also available by appointment.

My office is Wilkinson 143, but I'm not there very often. It is best to reach me via e-mail. Barring exceptional circumstances, I will respond to e-mails within 24 hours, and typically sooner.

Resources

There are no textbooks for this course. The primary resources we will be using are:

Dive Into Python, an online Python book. (abbreviated as "DIP" in course outline)

<http://diveintopython.org/toc/index.html>

© 2004, Mark Pilgrim

ArcGIS Desktop Help

Documentation at Python.org

Learning Outcomes

By the end of this course, students should be able to:

- ✓ Analyze a geoprocessing task and structure it as a sequential series of steps that can be expressed in a flow diagram or as "pseudocode"
- ✓ Use basic programming tools (iteration and conditional statements) and Python data structures (lists, dictionaries) to accomplish geoprocessing tasks.
- ✓ Explain the function of existing geoprocessing scripts.
- ✓ Write and document scripts that:
 - Run a series of geoprocessing tools on several data sets
 - Obtain and use information from attribute tables
 - Add and calculate fields in attribute tables
 - Create and edit geographic data sets
 - Read and write information between geographic data sets and text files

Course Outline & Schedule

Week 1, Sept. 28 - Introduction and Model Builder Refresher

Introduction to course, instructor, each other

Geoprocessing basics

Model Builder

Introduction to data set

→ Exercise 1: Create a model in Model Builder

Week 2, Oct. 5 - Programming Basics and Introduction to Python

→ Exercise 1 due

Programming basics: Variables, data types, statements, functions and arguments

IDLE: Python's interactive programming window

Basic Python syntax

Week 3, Oct. 12 - Lists, Dictionaries, Iteration, and Conditionals

◇ DIP sections 3.1, 3.2, 6.3

Introduction to Dive Into Python

Lists and Dictionaries

Iteration, especially over lists (for, while)

Conditional statements (if, else)

→ Exercise 2: Lists and dictionaries

Week 4, Oct. 19 - Object-Oriented Programming Concepts

◇ DIP section 6.1

Objects, attributes, and methods

Modules and functions

Debugging and error handling

Week 5, Oct. 26 - Working with Geoprocessing Tools in Scripts

→ Exercise 2 due

Building strings

Working with workspaces

Calling geoprocessing tools in scripts

Listing data

→ Exercise 3: Apply geoprocessing tools to multiple feature classes

Week 6, Nov. 2 - Cursors and Additional Tools for Strings

→ Exercise 3 due

◇ DIP sections 3.5, 3.7, parts of 7.2

Cursors, GetValue and SetValue

More on strings: split, formatting

Real expressions

→ Exercise 4: Working with tables using cursors

Week 7, Nov. 9 - Working with Geometries

→ Exercise 4 due

◇ DIP sections 3.6, 3.7

Tools for working with geometries: point, array and spatial reference objects

List comprehensions

Join method

→ Exercise 5: Working with geometries using cursors

Week 8, Nov. 16 - Working with Text Files

→ Exercise 5 due

Useful modules: csv, fileinput

Read/write geometries

→ Exercise 6: Working with text files

Week 9, Nov. 23 - Working with Rasters, Guest lecture

→ Exercise 5 due

Week, 10, Nov. 30 - Free/flex time

(No final exam.)

Course Policies

Attendance: Attendance is expected at all class meetings. Attendance is particularly important since there is no textbook, so lectures are the primary means of obtaining the information needed to complete exercises. If you will be unable to attend a session (e.g. due to conference attendance or field work), let me know in advance and be sure to get notes from someone.

Individual/collaborative work: Each student is expected to do the final work on the exercises individually. You should each write your own scripts, and comments in particular should be in your own words. However, you are encouraged to work together to discuss general approaches to the geoprocessing problems and get help from each other.

Turning in assignments: Exercises are due by midnight on their due date, as noted in the schedule above. Exercises should be submitted via e-mail, to kuglert@geo.oregonstate.edu. The subject line should be:

"Geo599, exercise X, your name"

Attached files should be named according to the following convention:

Geo599exX_LastName_part (e.g. Geo599ex2_Kugler_script.py)

The parts of each exercise to be turned in will be specified in the exercise descriptions.

Late assignments: Ten points per day late will be deducted from late assignments. Exercises more than a week late will not be accepted. (However, note that this is the first time this course has been taught and I'm not sure how long the exercises will take you. Please let me know if you are having difficulty completing them on time, and if there seems to be a general need for more time, I will consider revising the schedule.)

Grading

Grading for the course will be based on 6 exercises. There are no exams.

Exercise 1: 50 points	Course grading scale:	
Exercise 2: 70 points	415-450 = A	340-354 = C+
Exercise 3: 90 points	400-414 = A-	325-339 = C
Exercise 4: 80 points	385-399 = B+	310-324 = C-
Exercise 5: 80 points	370-384 = B	<310 = D
Exercise 6: 80 points	355-369 = B-	
TOTAL: 450 points		

Most of the exercises consist of adding comments to an existing script and then writing your own script. The purpose of adding comments to an existing script is to demonstrate your understanding of what the script accomplishes as a whole, and what each part of the script does. The most important aspect of your own scripts is that they accomplish the indicated geoprocessing task. They should also be well documented and reasonably efficient and robust. Points will be assigned as follows:

Commenting – 30 points

- ✓ Header comment describing the overall function of the script, 5 points
- ✓ Comments sufficient to describe each part of the script, 5 points
- ✓ Comments demonstrate understanding of the script, 20 points

Script writing – 50 points

- ✓ Script accomplishes intended purpose, as demonstrated by output, 30 points
- ✓ Documentation via comments allows easy comprehension of script, 10 points
- ✓ Script is efficient and robust, 10 points
 - Appropriate use of iteration, approach not overly complicated
 - Appropriate use of print statements to track progress
 - Appropriate use of try...except structure

Other tasks (exercises 1, 2a, 3a) – Points as indicated in exercise descriptions