

What I did with Python over the Summer

What can you actually do with Python and arcpy?



Why Python?



- VBA going away
- Open source
- Easy to learn?
- Growing popularity



Rich help pages



ArcGIS Help 10.2, 10.2.1, and 10.2.2

Resource Center

- ArcPy
 - Introduction
 - ArcPy functions
 - ArcPy classes
 - Alphabetical list of ArcPy classes
 - Cursor
 - Exceptions
 - FeatureSets/RecordSets
 - Fields
 - General
 - Geometry
 - Geometry
 - Multipoint
 - Point
 - PointGeometry
 - Polygon
 - Polyline
 - Geostatistical Analyst
 - Graphing
 - Parameter
 - Data Access module
 - What is the data access module?
 - Classes
 - Functions
 - Mapping module
 - Network Analyst module
 - Spatial Analyst module
 - Time module
- Environment settings
- Tool reference
- Tool errors and warnings

Geodata
Services

Geometry (arcpy)

Desktop » Geoprocessing » ArcPy » ArcPy classes » Geometry

Summary

Geometry objects define a...

Discussion

In many geoprocessing workflows, you use a feature class with cursors to iterate through the data from scratch using `Geometry`, `Point`, `Polygon`, or `Polyline`.

Syntax

Geometry (geometry, inputs, spatial_reference, has_z, has_m)

Parameter
geometry
inputs
spatial_reference
has_z
has_m

Properties

Property
JSON (Read Only)

- Help is directed toward Python
- Examples on nearly every page
- Detailed documentation for properties and methods

ArcPy



Esri's core Python API

- Nearly everything in the Data Management toolbox
- Additional classes and methods
- `arcpy.da` and `arcpy.mapping`

arcpy.mapping .mxd's & the TOC

- Can reach into a project file, and modify layers
- Can access the currently running project
 - Make layers
 - Modify layers
 - Remove layers

arcpy.da



The Data Access Module

- Probably the most important functionality of arcpy
- Cursors
 - SearchCursor - Read only
 - InsertCursor - Write one new record
 - UpdateCursor - Full access to read & write

Cursors



Control your data

```
arcpy.da.SearchCursor( FC, fields, whereClause)
```

- Restricting your data (that's a good thing)
 - Restrict your columns with the 'fields' parameter
 - Restrict your rows with the 'whereClause' parameter
- Full access to returned rows and cell values
 - Field Calculator on steroids

Geometry



Get to the point

- All Feature classes have a shape field
- arcpy has classes and methods to get spatial data
 - Geometry tokens for cursors “SHAPE@XY”
 - Classes
 - Geometry
 - Point
 - Polyline
 - Polygon

Python is Object-Oriented



- Nearly everything in Python is an object
 - Strings have methods
 - Lists and tuples are powerful, out of the box data structures which ArcPy takes advantage of
 - Dictionaries are less used, but highly useful.

Classes



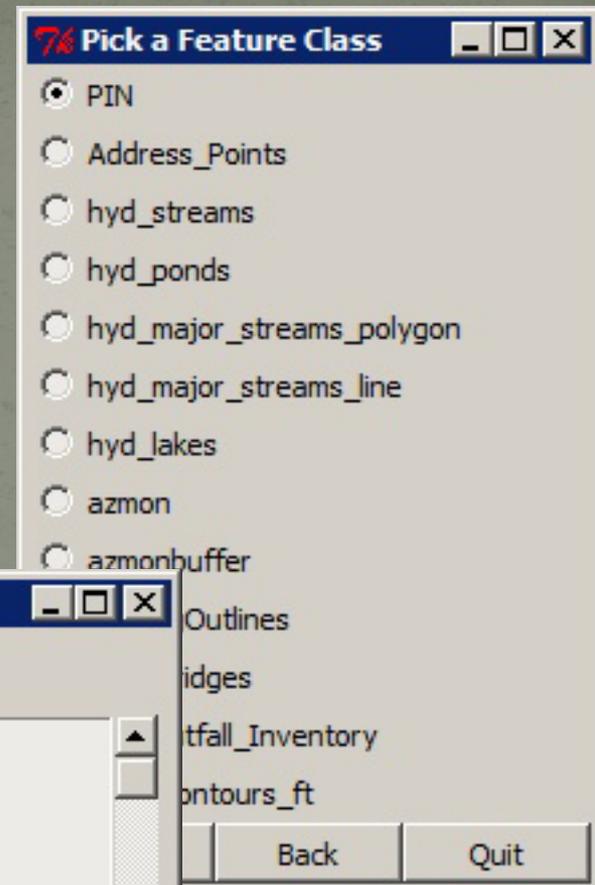
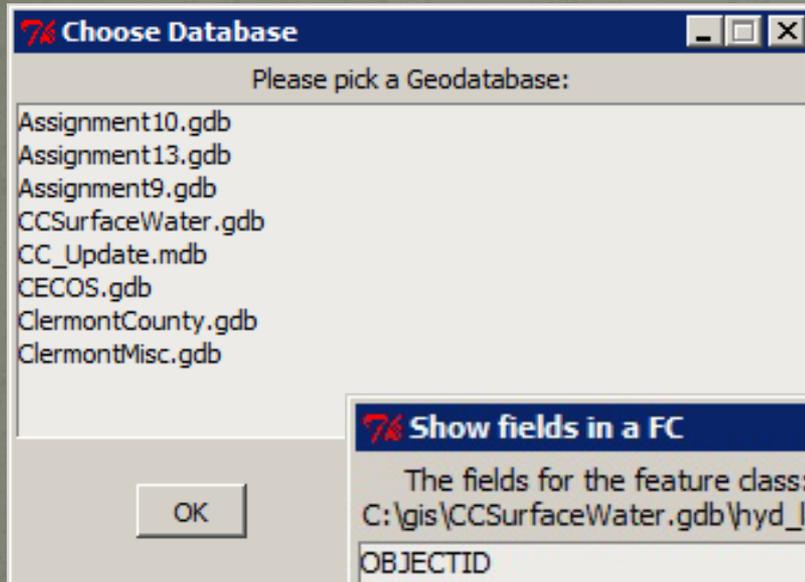
Easily made, very useful

```
class StreamTree:
    StreamCount          = 0
    ProblemStreamCount  = 0

    def __init__(self, num):
        self.StreamNum    = num
        self.LeftTrib     = None
        self.RightTrib    = None
        self.ProblemState = "None"
        assert self.ProblemState in ("ThreeWay", "OneChild", "None", "Leaf")
        StreamTree.StreamCount += 1

    @staticmethod
    def getEndPts():
        StreamEndPtFields = [ "streamNum", "SHAPE@" ]
        with arcpy.da.SearchCursor(StreamTree.streamFC, StreamEndPtFields) as streamCursor:
            for stream in streamCursor:
                StreamTree.endPointLookup[stream[0]] = ( s[1].lastPoint.X, s[1].lastPoint.Y )
```

Tk

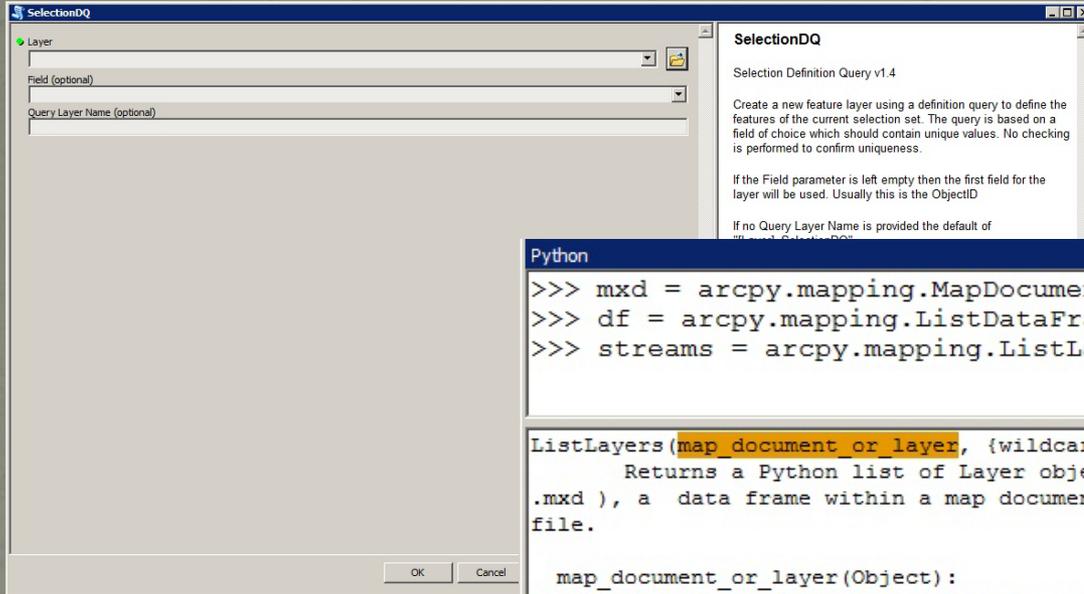


Myriad other modules



- All of the amenities of a fully developed language
 - Full OS access
 - csv, dbfPy
 - smtplib
 - urllib
 - math
 - NumPy, SciPy
 - Regular expressions

ArcMap, ArcCatalog, or stand-alone



```
Python
```

```
>>> mxd = arcpy.mapping.MapDocument("CURRENT")
>>> df = arcpy.mapping.ListDataFrames(mxd)
>>> streams = arcpy.mapping.ListLayers(
```

ListLayers(**map_document_or_layer**, {wildcard}, {data_frame})
Returns a Python list of Layer objects that exist within a map document (.mxd), a data frame within a map document, or layers within a layer (.lyr) file.

map_document_or_layer(Object):
A variable that references a MapDocument or Layer object.

wildcard(String):
A combination of asterisks (*) and characters can be used to help limit the results.

data_frame(DataFrame):
A variable that references a DataFrame object.

ArcToolbox
Identity
Python
Search

Command Prompt

```
C:\gis>python updategis.py
```

Debugging



Well, it depends....

The screenshot displays the Pyscripter IDE interface. The main window shows a Python script named 'StreamTree.py' with the following code:

```
1 #-----
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:    William F. Mellman
6 #
7 # Created:   2014-12-04
8 # Copyright: (c) William F. Mellman (bmellman) 2014
9 # Licence:   GNU General Public License
10 #-----
11
12 import arcpy
13 from PyModules import c360ify
14
15 MeanLines = "c:/gis/Assignment13.gdb/MeanLines"
16
17
18 @staticmethod
19 def getCompassA(num):
20     MeanLineFields = ["CompassA"]
21     whereClause = "StreamNum = " + str(num)
22     with arcpy.da.SearchCursor(MeanLines, MeanLineFields, whereClause) as MeanLineCursor:
23         return MeanLineCursor.next()[0]
24
25
26
27 class StreamTree:
28     StreamCount = 0 # Total number of streams, incremented in init
29     ProblemStreamCount = 0 # Total number of streams with non-None problem state
30     endPointLookup = {} # end points of all lines in streamsFC
31     streamFC = "c:/gis/Assignment13.gdb/streams"
32
33
34     def __init__(self, num):
35         self.StreamNum = num
36         self.LeftTrib = None
37         self.RightTrib = None
38         self.StreamCount = 0
39         self.ProblemState = "None"
40         assert self.ProblemState in ("FourWay", "OneChild", "None")
41         StreamTree.StreamCount += 1
42
43
44     @staticmethod
45     def getEndPts():
46         StreamEndPtFields = ["StreamNum", "SHAPE"]
47         with arcpy.da.SearchCursor(StreamTree.streamFC, StreamEndPtFields) as streamCursor:
48             for stream in streamCursor:
49                 StreamTree.endPointLookup[stream[0]] = (stream[1].lastPoint.X, stream[1].last
50
51
52     @staticmethod
53     def getRightLeft(trunk, A, B):
54         AngleA = getCompassA(A)
55
```

The 'Variables' window shows the following table:

Name	Type	Value	Namespace
globals	dict	{MeanLines: 'c:/gis/Assignment13.gdb/MeanLines'}	Frame(Function: "__init__" of module: "StreamTree" at line 38)
locals	dict	{num: 210, 'self': <__main__.StreamTree instance at 0x118158B80>}	
num	int	210	
self	StreamTree	<__main__.StreamTree instance at 0x118158B80>	
buildBranch	function	<function buildBranch at 0x118158B80>	
buildTree	function	<function buildTree at 0x118158B80>	
endPointLookup	dict	{46: (1476756.5279999971, 424034.6...}	
getChildList	function	<function getChildList at 0x118158B80>	
getEndPts	function	<function getEndPts at 0x118158B80>	
getRightLeft	function	<function getRightLeft at 0x118158B80>	
LeftTrib	NoneType	None	
ProblemStreamC...	int	0	
RightTrib	NoneType	None	
StreamCount	int	7	
streamFC	str	'c:/gis/Assignment13.gdb/streams'	
StreamNum	int	210	
doc	None	None	
init	instancemethod	<bound method StreamTree.__init_...	
module	str	'__main__'	

The 'Python Interpreter' window shows the following code:

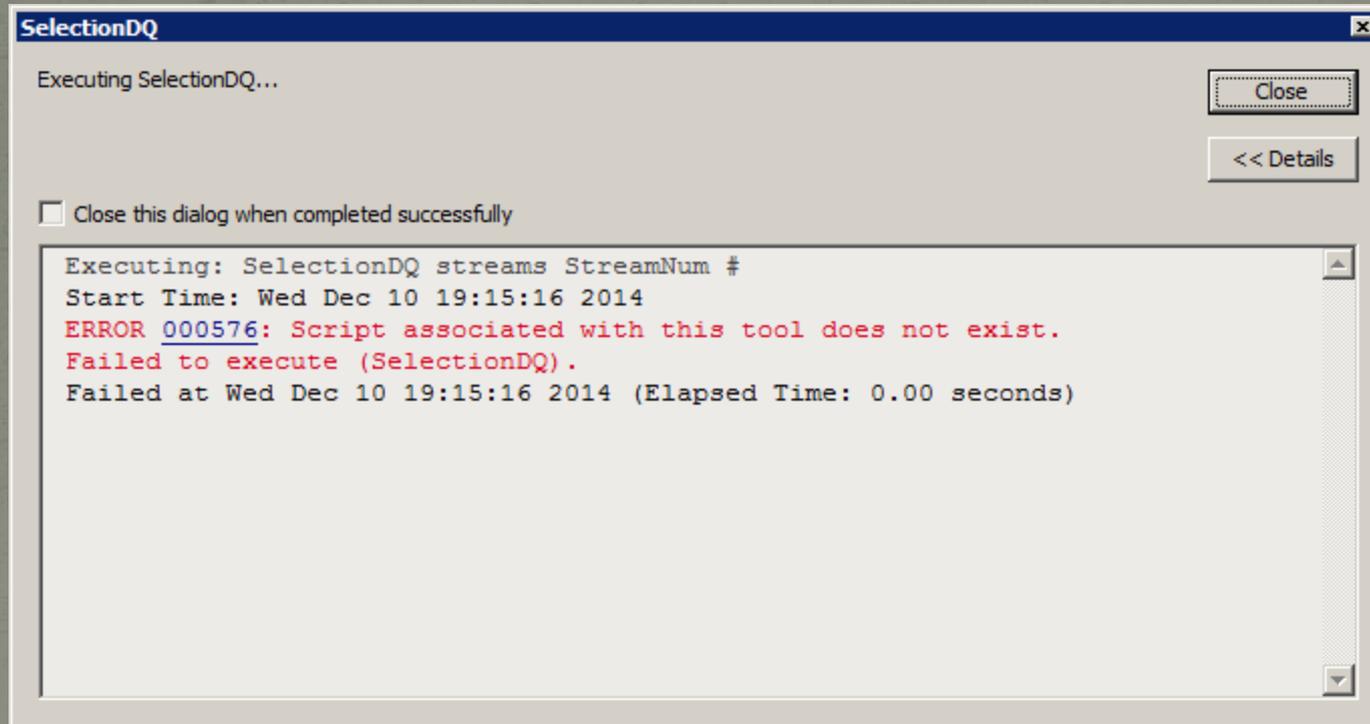
```
Python Interpreter
*** Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32 ***
>>>
[DBG]>>>
245
217 Four Way
215
213
```

Life is Good with Pyscripter!

But, in ArcMap....



Ouch!



No viable integrated debugging environment,
Can you say “print”?

Try & Except



```
• 1 try:  
• 2     x = 1/0  
• 3  
• 4 except ZeroDivisionError:  
• 5     print "Don't divide by zero!"
```

```
Python Interpreter  
*** Python 2.7.5 (default  
(Intel)] on win32. ***  
>>>  
[Dbg]>>>  
Don't divide by zero!  
  
>>> |
```

- Can check for several different kinds of exceptions
- Compound statement
- Nested exceptions

Exceptions are a Class

- And Classes Inherit, so ...
- you can make your own exceptions:

```
Class BillsException(Exception):  
    #Do some stuff in here
```

```
#Main  
raise BillsException()
```



Questions are Free

Answers will cost you, but

there's a special going today.



Bill Mellman

Clermont County Office of Environmental Quality

Clermont County Water Resources

bmellman@clermontcountyohio.gov