



Automate Your GIS Using Python

Ryan Moore
Manager of Geographic Information Services
Rochester Public Utilities

What is Python and ArcPy?

- Python is a free, cross-platform, open-source programming language that is both powerful and easy to learn. It is widely used and supported. To learn more about Python, visit python.org.
- Python was introduced to the ArcGIS community with ArcGIS 9.0.
- ArcPy is a Python site package that provides a useful and productive way to perform geographic data analysis, data conversion, data management, and map automation with Python.
- ArcPy includes modules covering other areas of ArcGIS. ArcPy is supported by a series of modules, including a [data access module](#) (arcpy.da), a [mapping module](#) (**arcpy.mapping**), an [ArcGIS Spatial Analyst extension module](#) (**arcpy.sa**), and an [ArcGIS Network Analyst extension module](#) (arcpy.na).

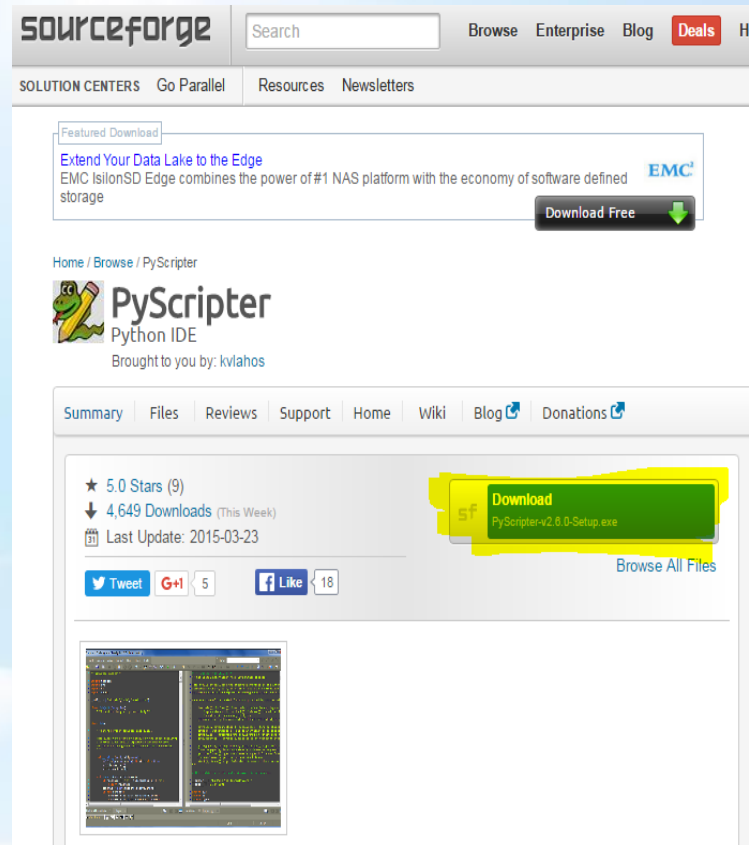
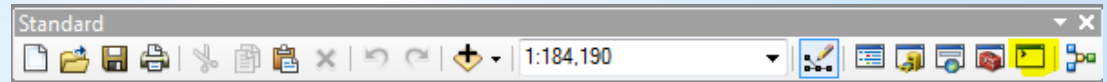
Source

<http://desktop.arcgis.com/en/arcmap/latest/analyze/python/what-is-python-.htm>

<http://pro.arcgis.com/en/pro-app/arcpy/get-started/a-quick-tour-of-arcpy.htm>

How to work with ArcPy?

- ArcGIS's Python Command Window (Available in ArcMap and ArcGIS Pro)
- Develop Python scripts using an integrated development environment (IDE) such as PyScripter



Running a Simple ArcToolbox Tool from the Command Window

- To figure out the syntax search for the tool
- Right click on the tool and select Help
- Scroll Down to bottom for sample code for the Python Window and stand alone script
- **Tip:** You can drag datasets and or workspaces from the ArcCatalog window to the Python window to get the file paths

Code Sample

Buffer example (Python window)

The following Python window script demonstrates how to use the Buffer tool.

```
import arcpy
arcpy.env.workspace = "C:/data"
arcpy.Buffer_analysis("roads", "C:/output/majorroadsBuffered", "100 Feet", "F
```

Buffer example (stand-alone script)

Find areas of suitable vegetation that exclude areas heavily impacted by major roads:

```
# Name: Buffer.py
# Description: Find areas of suitable vegetation which exclude areas heavil

# import system modules
import arcpy
from arcpy import env

# Set environment settings
env.workspace = "C:/data/Habitat_Analysis.gdb"

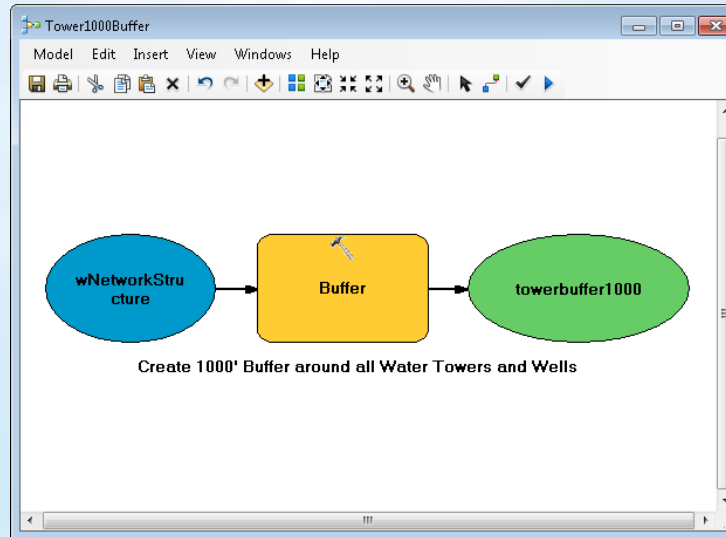
# Select suitable vegetation patches from all vegetation
veg = "vegtype"
suitableVeg = "C:/output/Output.gdb/suitable_vegetation"
whereClause = "HABITAT = 1"
arcpy.Select_analysis(veg, suitableVeg, whereClause)

# Buffer areas of impact around major roads
roads = "majorroads"
roadsBuffer = "C:/output/Output.gdb/buffer_output"
distanceField = "Distance"
sideType = "FULL"
endType = "ROUND"
dissolveType = "LIST"
dissolveField = "Distance"
arcpy.Buffer_analysis(roads, roadsBuffer, distanceField, sideType, endType,

# Erase areas of impact around major roads from the suitable vegetation pat
eraseOutput = "C:/output/Output.gdb/suitable_vegetation_minus_roads"
xyTol = "1 Meters"
arcpy.Erase_analysis(suitableVeg, roadsBuffer, eraseOutput, xyTol)
```

Building an ArcPy script from a model

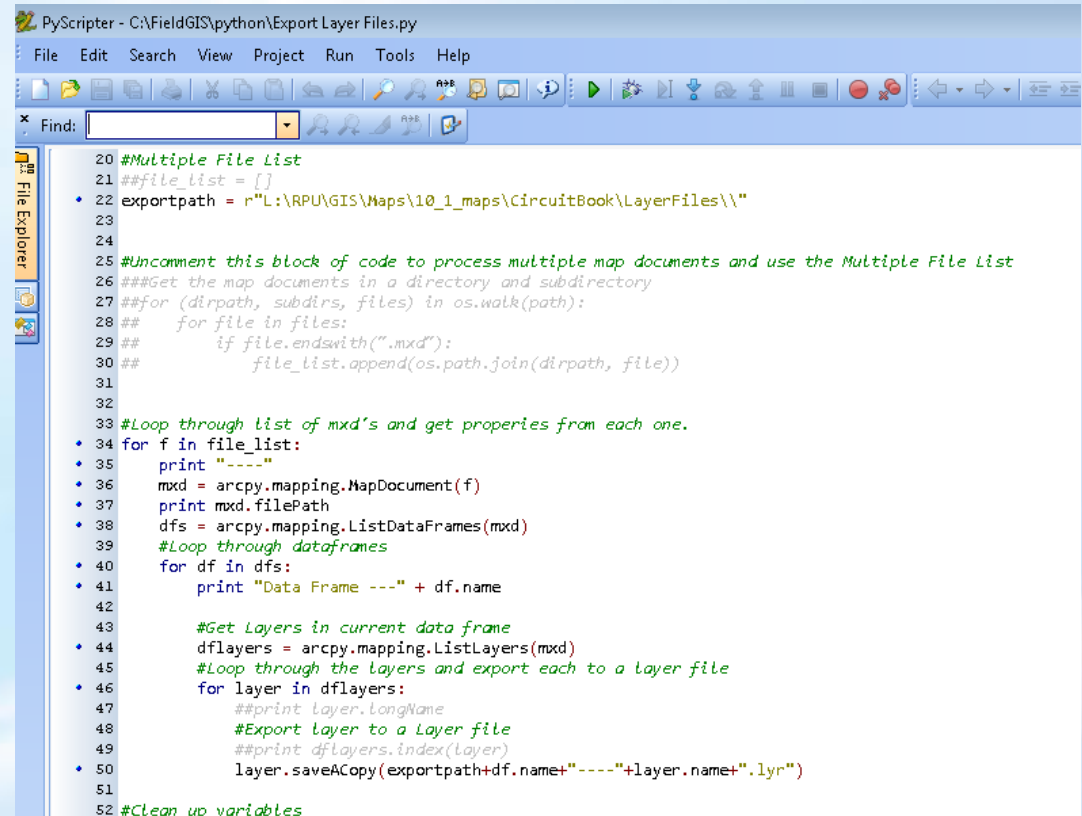
- Create the model in Model Builder
- Save the Model
- Click the Model – Export – To Python Scrip
- **Tip:** The script won't run “as-is” if you have any variables in the model that were created by dragging layers from the Table of Contents.



```
1 #-*- coding: utf-8 -*-
2 #
3 # tower1000Buffer_fullpath.py
4 # Created on: 2016-02-24 21:42:25.00000
5 # (generated by ArcGIS/ModelBuilder)
6 # Description:
7 #
8
9 # Import arcpy module
10 import arcpy
11
12
13 # Local variables:
14 wNetworkStructure = "C:\\FieldGIS\\rpugis_gdb\\rpugis.gdb\\Water_Distribution_Network\\wNetworkStructure"
15 towerbuffer1000 = "C:\\FieldGIS\\python\\pythondemo.gdb\\towerbuffer1000"
16
17 # Process: Buffer
18 arcpy.Buffer_analysis(wNetworkStructure, towerbuffer1000, "1000 Feet", "FULL", "ROUND", "NONE", "")
19
20
```


Building a Script in PyScripter

- Import the ArcPy and other necessary modules
- Keep in mind that Python is case sensitive
- Comments can be made starting a line with a # sign
- Indentation is used to organize blocks of code (Ex. If and For statements)
- Make use of the Debug tools and set breakpoints
- Utilize the IDE windows under the View menu Pulldown



```
PyScripter - C:\FieldGIS\python\Export Layer Files.py
File Edit Search View Project Run Tools Help
Find:

20 #Multiple File List
21 ##file_list = []
22 exportpath = r"L:\RPU\GIS\Maps\10_1_maps\CircuitBook\LayerFiles\\"
23
24
25 #Uncomment this block of code to process multiple map documents and use the Multiple File List
26 ##Get the map documents in a directory and subdirectory
27 ##for (dirpath, subdirs, files) in os.walk(path):
28 ##     for file in files:
29 ##         if file.endswith(".mxd"):
30 ##             file_list.append(os.path.join(dirpath, file))
31
32
33 #Loop through list of mxd's and get properties from each one.
34 for f in file_list:
35     print "----"
36     mxd = arcpy.mapping.MapDocument(f)
37     print mxd.filePath
38     dfs = arcpy.mapping.ListDataFrames(mxd)
39     #Loop through dataframes
40     for df in dfs:
41         print "Data Frame ----" + df.name
42
43         #Get Layers in current data frame
44         dflayers = arcpy.mapping.ListLayers(mxd)
45         #Loop through the layers and export each to a layer file
46         for layer in dflayers:
47             ##print layer.longName
48             #Export layer to a Layer file
49             ##print dflayers.index(layer)
50             layer.saveACopy(exportpath+df.name+"----"+layer.name+".lyr")
51
52 #Clean up variables
```

Python Syntax for Special Characters and Comparisons

Escape Sequences

This all of the escape sequences Python supports. You may not use many of these Try them out in some strings to see if you can make them work.

ESCAPE	WHAT IT DOES.
<code>\</code>	Backslash (\)
<code>'</code>	Single-quote (')
<code>"</code>	Double-quote (")
<code>\a</code>	ASCII bell (BEL)
<code>\b</code>	ASCII backspace (BS)
<code>\f</code>	ASCII formfeed (FF)
<code>\n</code>	ASCII linefeed (LF)
<code>\N{name}</code>	Character named name in the Unicode database (Unicode only)
<code>\r</code>	Carriage Return (CR)
<code>\t</code>	Horizontal Tab (TAB)
<code>\uxxxx</code>	Character with 16-bit hex value xxxx (Unicode only)
<code>\Uxxxxxxxx</code>	Character with 32-bit hex value xxxxxxxx (Unicode only)
<code>\v</code>	ASCII vertical tab (VT)
<code>\ooo</code>	Character with octal value ooo
<code>\xhh</code>	Character with hex value hh

This table summarizes the comparison operations:

Operation	Meaning	Notes
<code><</code>	strictly less than	
<code><=</code>	less than or equal	
<code>></code>	strictly greater than	
<code>>=</code>	greater than or equal	
<code>==</code>	equal	
<code>!=</code>	not equal	(1)
<code>is</code>	object identity	
<code>is not</code>	negated object identity	

Notes:

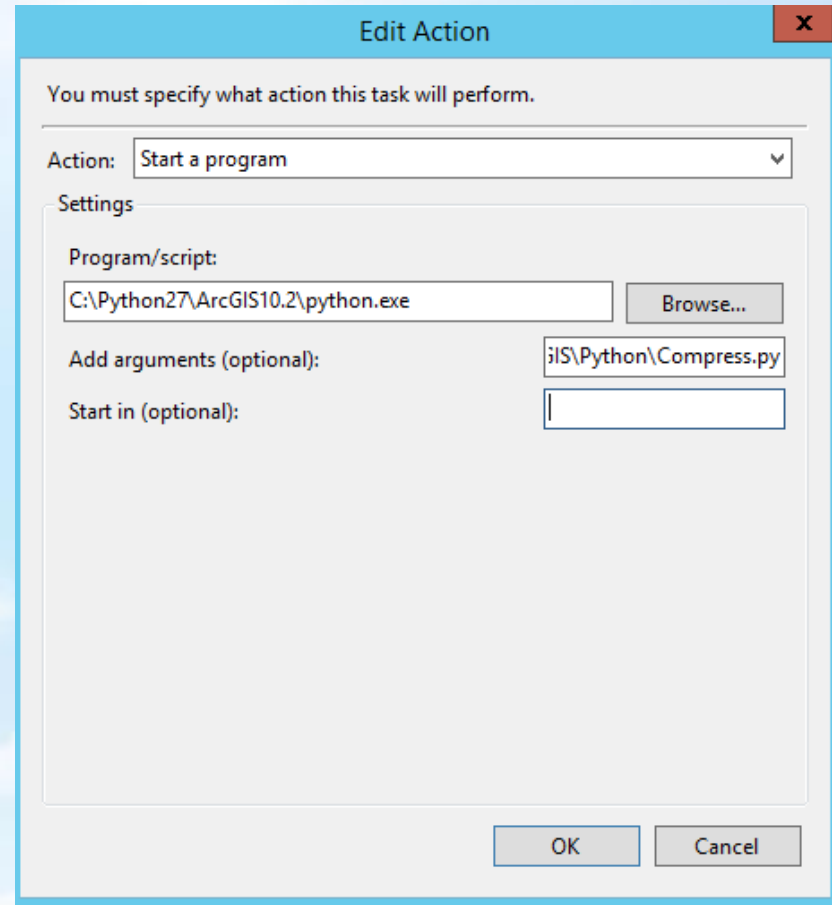
1. `!=` can also be written `<>`, but this is an obsolete

Scripts Used at RPU

- Geodatabase Maintenance Scripts
 - Reconcile Versions.py
 - Compress.py (Runs as a scheduled Task)
- Map document changes/exports
 - Export Layers.py
 - Update Definition Queries.py
 - Export map documents to PDF.py

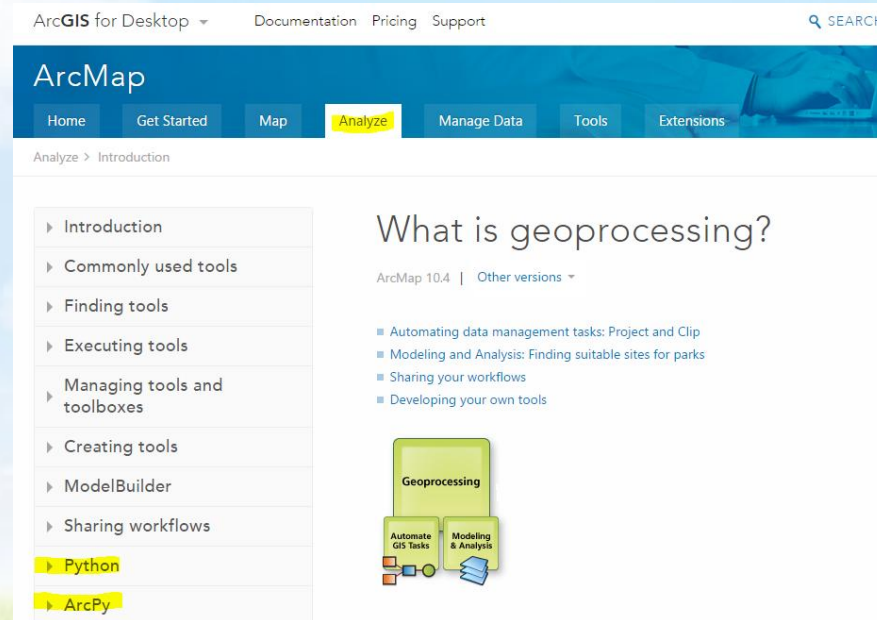
Running a Python Script using Task Scheduler

- Open Task Scheduler and create a new task
- Under the Action Tab create a new action “Start a program”
- Browse for your python.exe
- In the Add arguments text box put the path and filename of the python script you want to run



Where to find ArcPy Documentation and Examples?

- ArcToolbox Desktop Help – Geoprocessing Tools
- ArcGIS for Desktop online help (Click the Analyze Tab)
- <http://desktop.arcgis.com/en/arcmap/latest/analyze/main/what-is-geoprocessing.htm>



Additional Resources

Books

- [Programming ArcGIS with Python Cookbook - Second Edition Paperback – July 28, 2015](#)
- [Python Scripting for ArcGIS Pap/DVD Edition](#)

Online

- ESRI Virtual Campus
<http://www.esri.com/training/main/training-catalog>
- Learn Python the Hard Way
<http://learnpythonthehardway.org/book/>
- Python Documentation
<https://docs.python.org/2/contents.html>

The screenshot shows the Esri Training Catalog search results for the keyword 'python'. The page has a dark blue header with the Esri logo and navigation links: Industries, Products, Support & Services, About, and Community. Below the header is a 'Training' section with tabs for Main, Training Catalog (selected), My Training, News, Certification, and Common Questions. A user greeting 'Welcome, Ryan' is visible. The search results section is titled 'Training Catalog Search Results' and shows '5 training options found'. It includes a 'Sort by' dropdown set to 'Relevance' and a 'Showing 1 to 5 of 5 results' message. The results are listed in a table with columns for course title, format, duration, price, and ArcGIS version. The courses are: 'Python for Everyone', 'Basics of Python (for ArcGIS 10)', 'Python Scripting for Geoprocessing Workflows', 'Python Scripting for Map Automation', and 'Esri Technical Certification: Sample Questions for ArcGIS Desktop Developer Associate'. To the right of the results are filters for 'Current Search' (python) and 'Narrow Your Search' (Keyword and ArcGIS Version).

Course Title	Format	Duration	Price	ArcGIS Version
<u>Python for Everyone</u> ▶ Show Overview	Web Course	1 module (3 hours)	Free	10.1, 10.2, 10.3, 10.4
<u>Basics of Python (for ArcGIS 10)</u> ▶ Show Overview	Web Course	1 module (3 hours)	\$32 USD	10.0
<u>Python Scripting for Geoprocessing Workflows</u> ▶ Show Overview	Web Course	1 module (3 hours)	\$32 USD	10.0, 10.1, 10.2, 10.3, 10.4
<u>Python Scripting for Map Automation</u> ▶ Show Overview	Web Course	1 module (3 hours)	\$32 USD	10.0, 10.1, 10.2, 10.3, 10.4
<u>Esri Technical Certification: Sample Questions for ArcGIS Desktop Developer Associate</u> ▶ Show Overview	Web Course	1 module (60 minutes)	Free	10.1