Getting Started with Python

Carl Eichenberger

Agenda

- What is Python and why use it?
- Python basics
- The ArcPy module – Python’s window to ArcGIS
- Geoprocessing script tools
- Resources

What is Python?

"Python is an easy to learn, powerful language… (with) high level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing…make it an ideal language for scripting…in many areas and on most platforms."

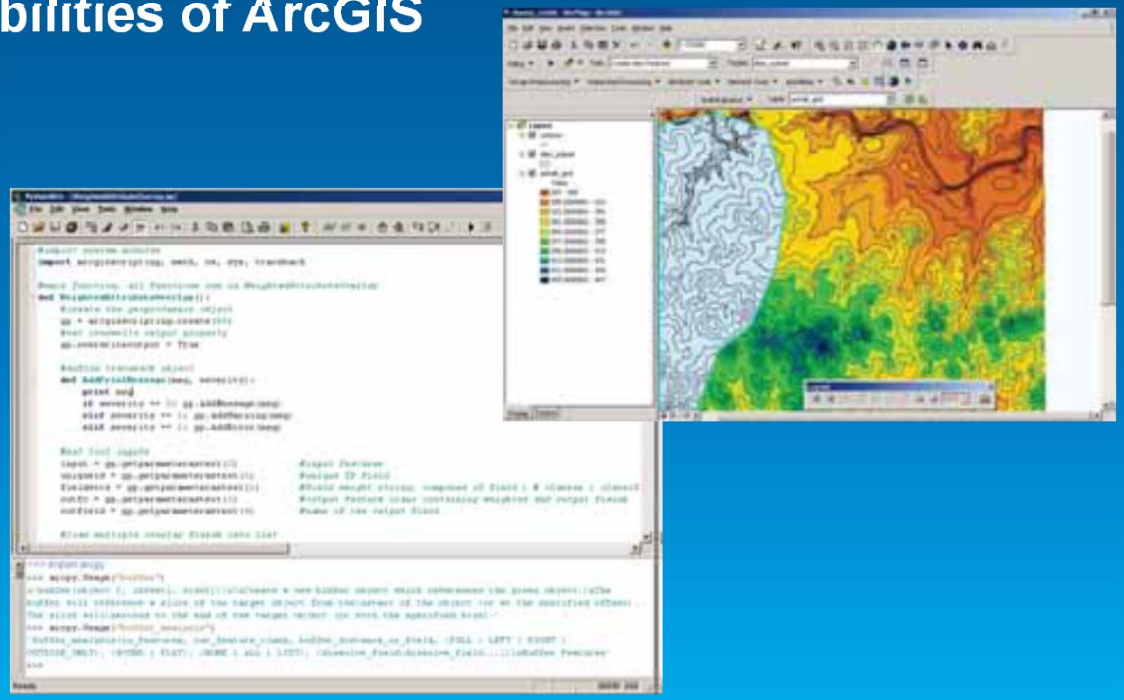
– python.org

Scripting language of ArcGIS

Free, cross-platform, easy to learn, widely useful, great community

Why use Python and ArcGIS?

- Automate repetitive tasks
- Make your own spatial analysis tools
- Add geoprocessing to web applications
- Extend the capabilities of ArcGIS



Python Basics

Where do I write Python code?

- **Python file is text with .py extension**
- **IDE like PyScripter, Wing IDE ($), PythonWin, IDLE**
- **Python window in ArcGIS**

Desired task	Python window	Python IDE
Execute a single line of code	✓	✓
Work directly with layers in ArcMap	✓	
Test geoprocessing workflows and easily verify results	✓	
Work with tools to enter, edit, and check syntax in your code		✓
Work in an Integrated Debugging Environment		✓

Python building blocks

Function: a defined piece of functionality that performs a specific task; requires arguments

Module: a Python file where functions live; import

Package: a collection of related modules

```
import math
math.sqrt(100) … 10
```

<u>Python Standard Library / Built-ins</u>

os, sys, math, datetime, urllib2

Demo

Writing Python code

- Python basics
- Python IDE
- ArcGIS Python window

```
23 arcpy.SetProgressorLabel("preparing to
24 time.sleep(5)
25
26 # select vegetation of desired type
27 select = arcpy.Select_analysis(veg, 'in
28     ' "VEG_TYPE" IN (\'Coastal Sage-Cha
29
30 # buffer by 1000 feet
31 road_buff = arcpy.Buffer_analysis(roads
32
33 # remove buffered roads from desired ar
34 erase = arcpy.Erase_analysis(select, ro
35
36 # find interesection of valid climate,
37 intersect = arcpy.Intersect_analysis([c
38
39 # dissolve all adjacent areas together
40 dissolve = arcpy.Dissolve_management(in
41
42 # add field and calculate are of contig
43 arcpy.AddField_management(dissolve, 'Ar
44 arcpy.CalculateField_management(dissolv
45
46 # select out areas that match all crite
47 arcpy.Select_analysis(dissolve, outfc,
48 arcpy.AddMessage('NUMBER OF NON CONTIGU
49
50 arcpy.ResetProgressor()
51 print ('FINISHED')
```

```
no4_suitability_analysis_Tool.py ×
thon Interpreter
*** Remote Python engine  is active ***

*** Remote Interpreter Reinitialized  ***

g]>>>
```

Call Stack | Variables | Watches | Breakpoints

The ArcPy Module

- Site package that adds ArcGIS functionality to Python
- Access to rich suite of geoprocessing tools
- Functions, classes and modules
  - Helper functions like ListFeatureClasses, Describe
  - Classes that can be used to create complex objects like SpatialReference, FieldMap
  - Modules that provide specialized functionality like Mapping, SpatialAnalyst, NetworkAnalyst, DataAccess

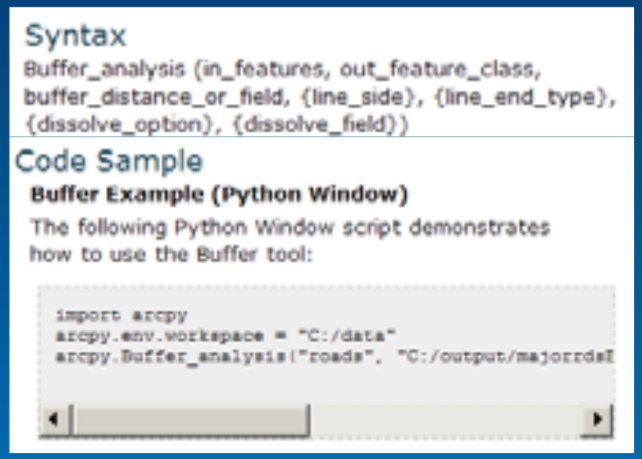
Run geoprocessing tools

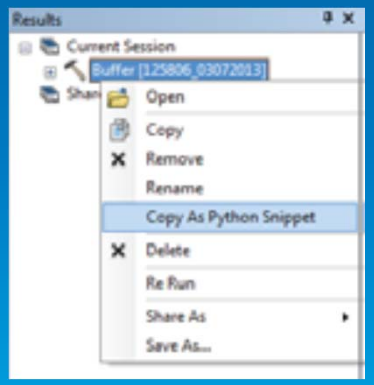
import arcpy

Tool syntax

arcpy.Buffer_analysis()

Enter input and output parameters





How do I use a specific tool?

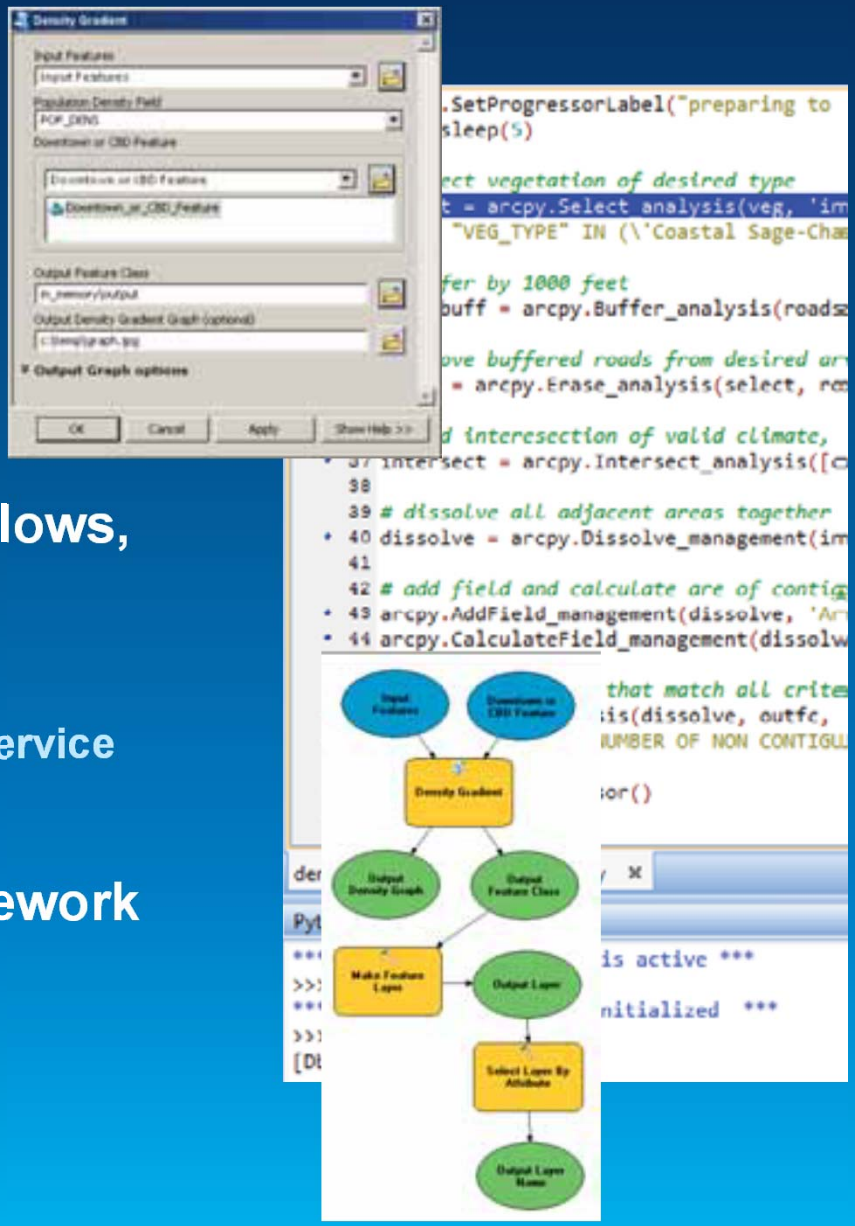
Tool help page

Copy as Python Snippet

help(arcpy.Buffer_analysis)

Export from Model Builder

Python script tools



Make your Python script into a geoprocessing tool

Great way to create and share workflows, extend ArcGIS

- More accessible than stand-alone script
- Share as a geoprocessing package or service

Integrated with geoprocessing framework

- Use it just like any other tool
- Works with map layers, selections

Demo

Geoprocessing script tool

```
23 arcpy.SetProgressorLabel("preparing to
24 time.sleep(5)
25
26 # select vegetation of desired type
27 select = arcpy.Select_analysis(veg, 'in
28     ' "VEG_TYPE" IN (\'Coastal Sage-Cha
29
30 # buffer by 1000 feet
31 road_buff = arcpy.Buffer_analysis(roads
32
33 # remove buffered roads from desired ar
34 erase = arcpy.Erase_analysis(select, ro
35
36 # find interesection of valid climate,
37 intersect = arcpy.Intersect_analysis([c
38
39 # dissolve all adjacent areas together
40 dissolve = arcpy.Dissolve_management(in
41
42 # add field and calculate are of contig
43 arcpy.AddField_management(dissolve, 'Ar
44 arcpy.CalculateField_management(dissolv
45
46 # select out areas that match all crite
47 arcpy.Select_analysis(dissolve, outfc,
48 arcpy.AddMessage('NUMBER OF NON CONTIGU
49
50 arcpy.ResetProgressor()
51 print ('FINISHED')
```

```
no4_suitability_analysis_Tool.py ×

thon Interpreter

*** Remote Python engine  is active ***

*** Remote Interpreter Reinitialized  ***

g]>>>

Call Stack | Variables | Watches | Breakpoints
```

Resources

resources.ArcGIS.com

arcpy.wordpress.com

GIS Stack Exchange, Stack Overflow

Python References

Python Scripting for ArcGIS

by Zandbergen (Esri Press)

GIS Tutorial for Python Scripting

by Allen (Esri Press)

Learning Python

by Lutz

The Python Standard Library by Example

by Hellmann

python.org

esri

Understanding our world.