Python: An Introduction

David Wynne, Jon Bodamer

Python: An Introduction

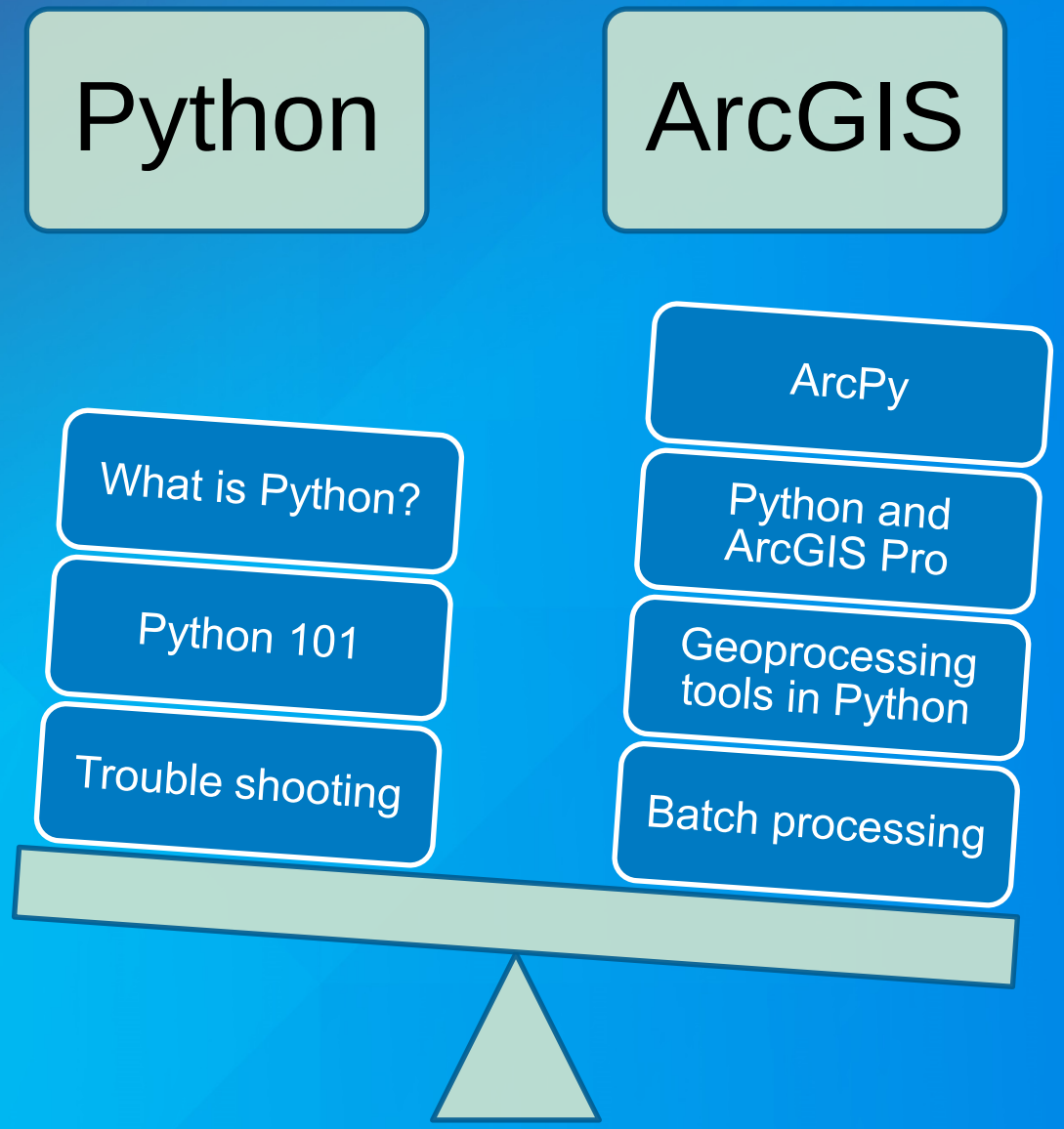
Beginner

Tuesday, 21 Jul 2015, 8:30am - 9:45am
Location: Ballroom 06 B

Thursday, 23 Jul 2015, 8:30am - 9:45am
Location: Ballroom 06 B

Python is a powerful and easy to learn open-source programming language that is widely used and supported within ArcGIS. Come learn about the foundations of Python, how to automate your workflows, and about ArcPy, the entry point for accessing the built-in functionality of ArcGIS through Python.

Agenda


Python
ArcGIS
What is Python?
Python 101
Trouble shooting
ArcPy
Python and ArcGIS Pro
Geoprocessing tools in Python
Batch processing

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**

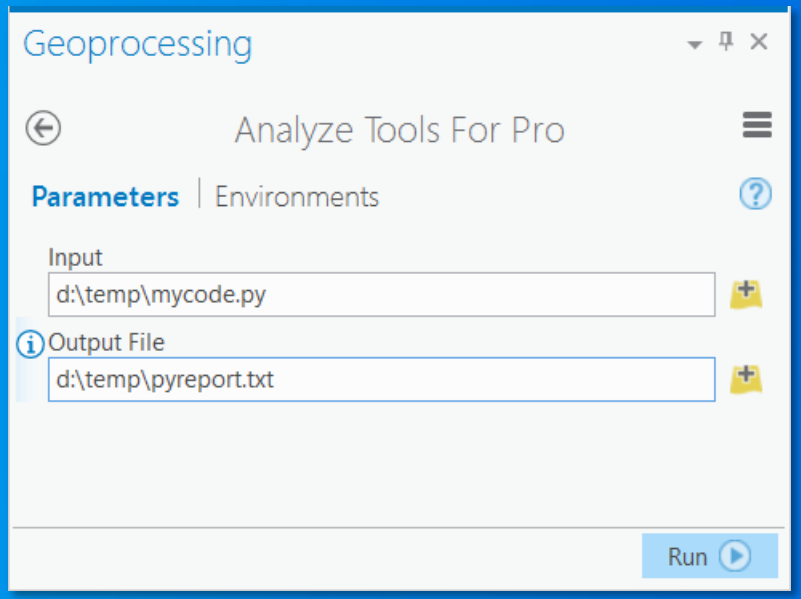
- **Scripting language of ArcGIS**
- **Free, cross-platform, easy to learn, widely useful, great community**

Why use Python and ArcGIS?

- Automate repetitive tasks
- Make your own geoprocessing tools
- Add geoprocessing to web applications
- Customize Desktop apps
- Extend the capabilities of ArcGIS

ArcGIS Pro and Python

- Python continues to be an important language for ArcGIS Pro
- Key differences
  - Most but not all the same geoprocessing tools
  - mapping changes (arcpy.mapping → arcpy.mp)
  - Pro uses 3.4 (other products uses 2.7)
- Can write Python code that will work in both
  - Analyze Tools For Pro (2to3)
  - http://esriurl.com/PythonProMigration



Python 101 - IDEs

- Python code is written in .py files
  - Or an interactive Python prompt (Python command line, Python window in ArcGIS)
- Where do I write Python code?
  - IDEs such as PyScripter, Wing IDE, PyCharm, Python IDLE, …
- How do I run?
  - Double-click .py, from command line, IDEs, ArcGIS

Python 101 - Comments

```
# Text with a leading '#' is a comment.


# Sometimes comments are descriptive and will wrap
# over multiple lines. We call these block comments


this = True  # inline comment


"""Documentation strings (docstrings) are important for your headers.
Also for functions, classes, methods.
"""
```

Python 101: Strings and string manipulation

- **Combining strings**

```
distance = 5
units = "miles"

buffer_distance = str(distance) + " " + units  # okay
buffer_distance = "{} {}".format(distance, units)  # better
```

- **Pathname manipulations**

```
import os

# Join path components
os.path.join(geodatabase, 'output_table')
# get directory
os.path.dirname(geodatabase)
```

Python 101 - Conditional logic

- Python has logic for testing conditions
  - if, elif, else statements
  - Colon at end of each condition
  - Indentation determines what is executed
  - == tests equality; other operators like >, <, !=

```
if value >= 0 and value <= 10:
    code = 1
elif value > 10:
    code = 2
else:
    raise Exception('Value cannot be negative')
```

Python 101 - iteration

- Python has for and while statements for looping
  - Colon at end of statement
  - Indentation determines what is executed

```
>>> feature_classes = ['roads', 'rivers', 'lakes']
>>> for fc in feature_classes:
...     print(fc)
...
roads
rivers
lakes
>>>
>>> x = 1
>>> while x < 5:
...     print(x)
...     x += 1
...
1
2
3
4
```

Python 101: libraries and import

- Python is ‘batteries included’
- Python has a broad collection of 3rd party libraries
- Can organize your own functionality and import it
  - Organize and re-use!

- Additional libraries are loaded using import

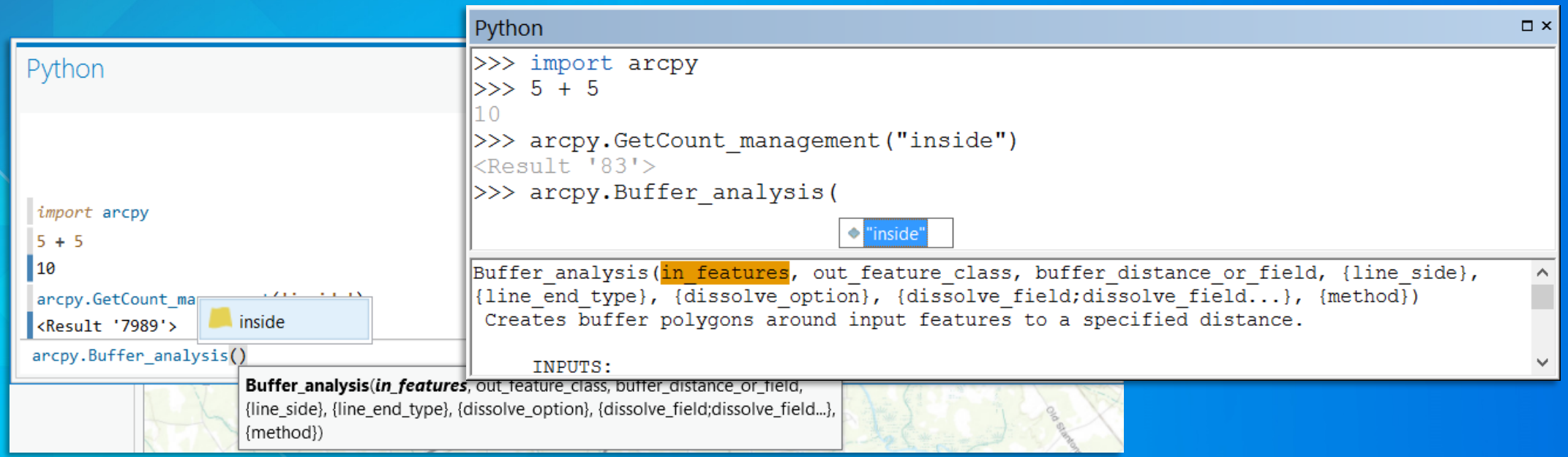
```
import arcpy

# standard libraries
import os

# 3rd party libraries
import numpy

# your own utilities
import myutils
```

Python basics

Demo

ArcGIS Python window

- Both Desktop and Pro have an embedded, interactive Python command line
- Access to Python and modules within ArcGIS applications
- Experiment with Python code interactively

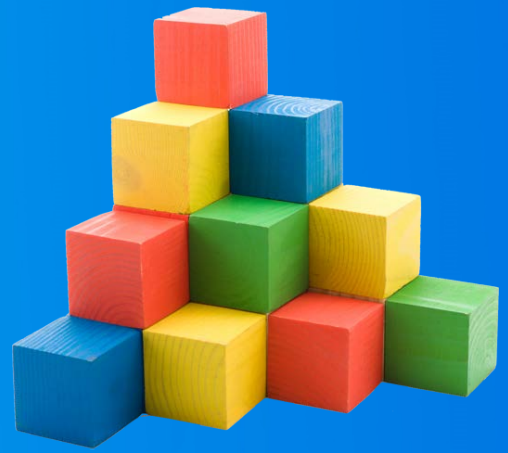


Python building blocks

- *Module*: a Python file where functions live; `import`
- Package: a collection of related modules

- *Function*: a defined piece of functionality that performs a specific task; requires arguments
- Class: a blueprint to create an object

- math.sqrt(100)



ArcPy

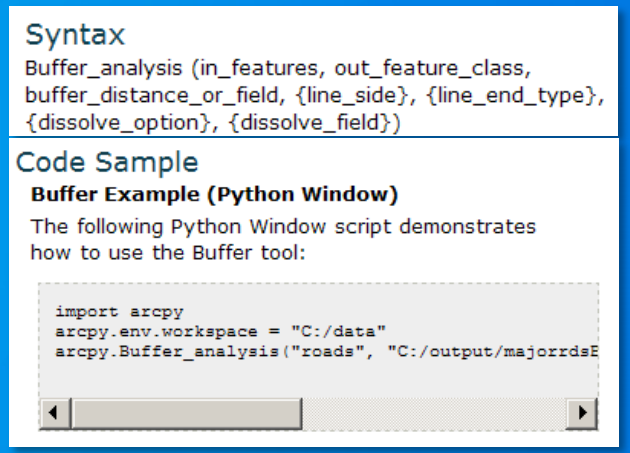
- Access point to ArcGIS functionality through Python
  - Desktop, Server, Engine, and Pro

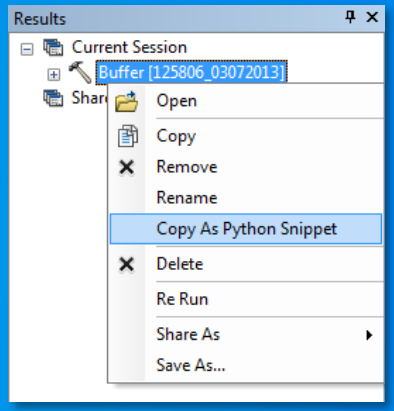
- Includes:

1. Geoprocessing tools
2. Functions like ListFeatureClasses, Describe
3. Classes that can be used to create objects like SpatialReference, FieldMap
4. Modules including Mapping (arcpy.mapping / arcpy.mp), Data access (arcpy.da), Spatial Analyst (arcpy.sa), Network Analyst (arcpy.na)

Run geoprocessing tools

- import arcpy
- Follow tool syntax
  - arcpy.toolname_toolboxalias(arguments)
    or
    arcpy.toolboxalias.toolname(arguments)
  - Include input and output parameters
- How do I use a specific tool?
  - Tool help page
  - Copy as Python Snippet
  - print(help(arcpy.Buffer_analysis))





Geoprocessing and Python

Demo

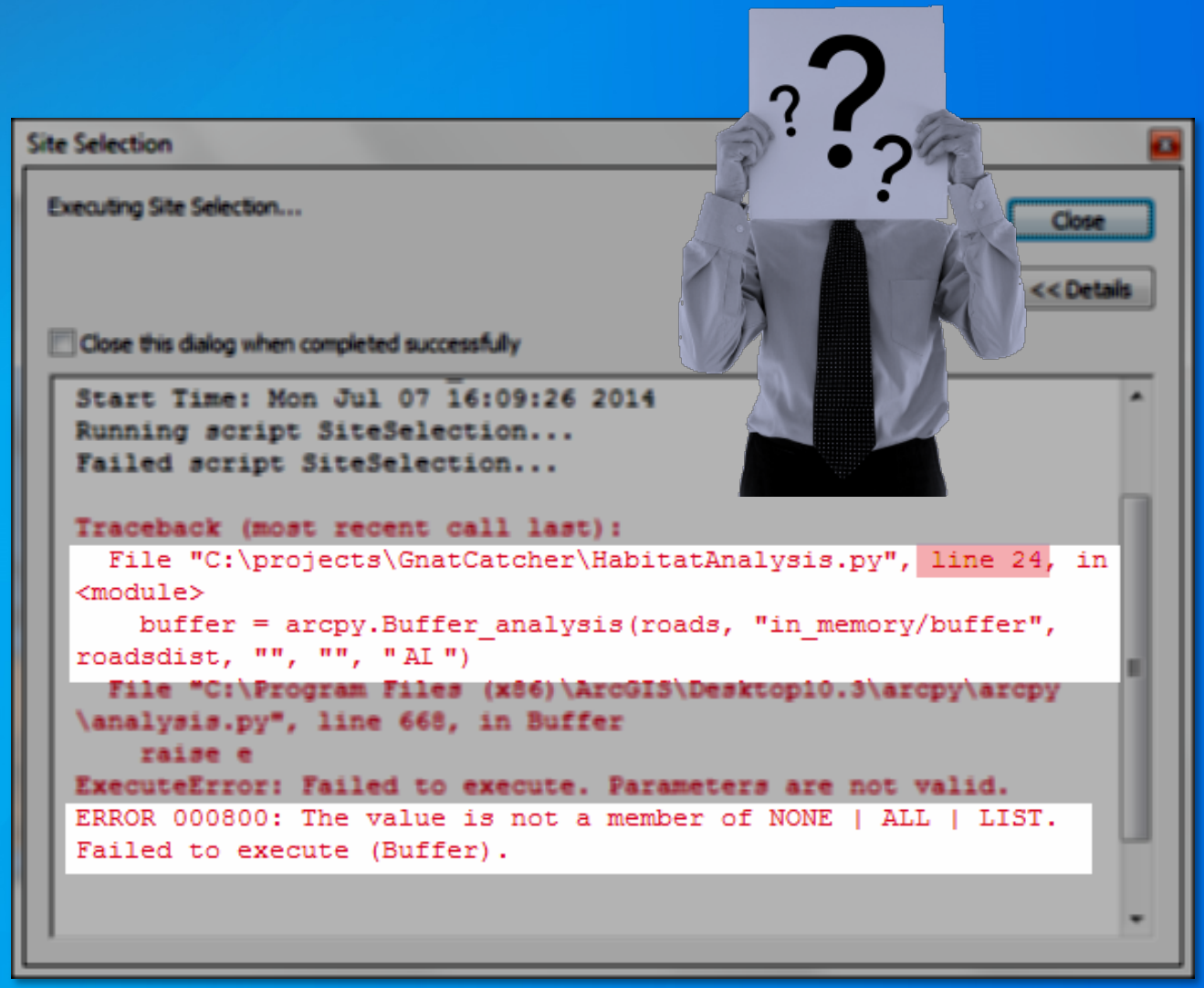
Geoprocessing environment settings

- Control the processing environment of the tools you run
  - See tool help for honored environments
- Productivity and code cleanup
- Environments are properties on arcpy.env (over 50)

```
arcpy.env.workspace = "c:/Data"

arcpy.env.extent = arcpy.Extent(0, 0, 100, 100)

arcpy.env.outputCoordinateSystem = 4326  # WKID
```

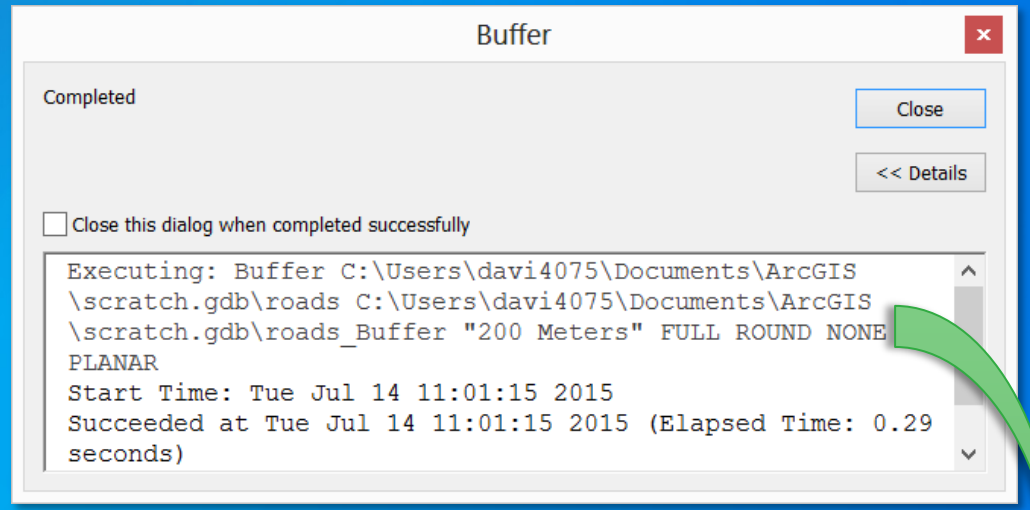
Troubleshooting

- Why do errors occur?
  - Incorrect tool use, typos, syntax, logic errors
- My script doesn't work?
  - Examine the messages
  - Use Python exception handling
  - Debug the script in an IDE



Geoprocessing tool messages

- Three types of messages
  - Informative, warning, error
- Displayed in ArcMap / Pro
  - Results
  - Messages window
  - Python window
- To access messages in Python
  - arcpy.GetMessages()



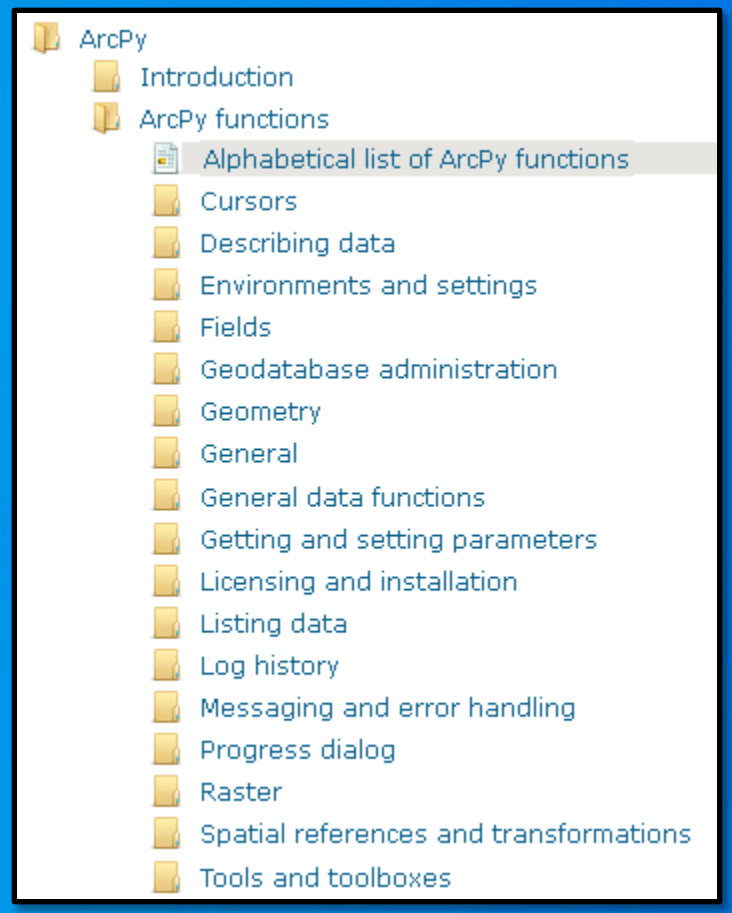
Python exception handling

- try / except statements
  - Try to do something, and if an exception occurs, do something else
- Many different exception types
- An arcpy.ExecuteError exception occurs when a geoprocessing tool fails

```
# Start try block
try:
    # more code

    arcpy.Buffer_analysis(in_features, output, buffer_distance)

    # more code

# If an tool error occurs
except arcpy.ExecuteError:
    print(arcpy.GetMessages())

# If a generic exception occurs
except Exception as err:
    print(err)
```

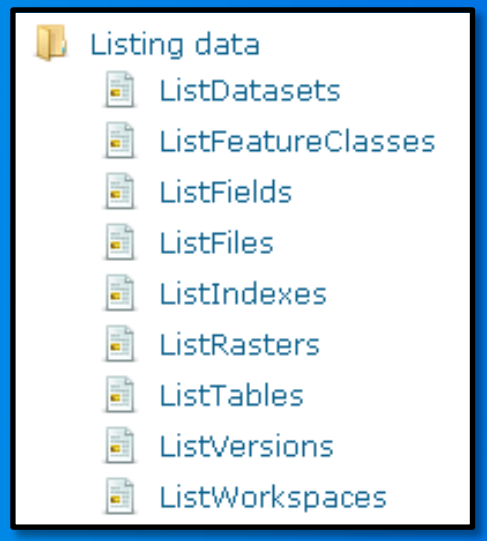
ArcPy functions

- An ArcPy function for many jobs
- Support geoprocessing workflows
- Enable automation of manual tasks



Batch processing

- Automating a process to run multiple times
  - Clip every feature class in a geodatabase to a common boundary
  - Calculate statistics for every raster in a folder
- List functions used in Python to perform batch processing
  - (also arcpy.da.Walk)



Batch processing (ListFeatureClasses)

```
# Set the workspace environment
arcpy.env.workspace = 'c:/data/FileGDB.gdb/fds'

# output workspace to write to
out_workspace = 'c:/data/output.gdb'

# Get a list of all feature classes
feature_classes = arcpy.ListFeatureClasses()

# Clip each feature classes
for fc in feature_classes:
    output = os.path.join(out_workspace, '{}_clip'.format(fc))
    arcpy.Clip_analysis(fc, boundary, output)
```

Getting data properties

- Describe functions reads data properties

- Returns an object with properties like:
  - Data type
  - Shape type
  - Spatial reference

```
# Describe a feature class
desc = arcpy.Describe("c:/Data/Roads.shp")

print(desc.shapeType)  # "Polyline"
```

Writing a Python script

Demo

Python at User Conference

- **http://esriurl.com/UC2015Python (54 Python sessions)**

	Tuesday	Wednesday	Thursday
8:30	• Python: An Introduction	• Python Map Automation: Introduction to…	• Python: An Introduction • Python: Raster Analysis
10:15	• Python: Beyond the Basics		• Python: Beyond the Basics • Python Map Automation: Introduction to …
11:30		• ArcGIS Pro: A Quick Tour of Python	
1:30	• Python: Building Geoprocessing tools		• Python: Building Geoprocessing tools • Advanced Map Automation with Python
3:15	• Python: Raster Analysis	• Advanced Map Automation with Python	

Resources

resources.ArcGIS.com

arcpy.wordpress.com

GIS Stack Exchange, Stack Overflow

www.esri.com/training

Python References

Python Scripting for ArcGIS
by Zandbergen (Esri Press)

GIS Tutorial for Python Scripting
by Allen (Esri Press)

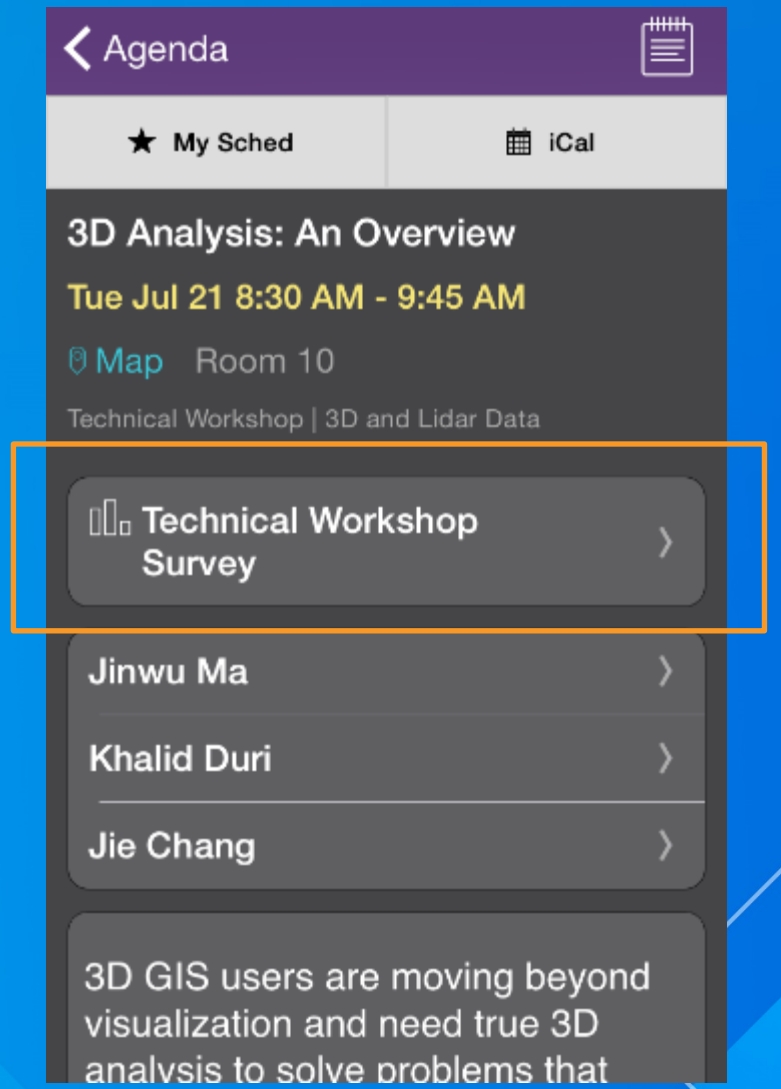
Learning Python by Lutz

The Python Standard Library by Example
by Hellmann

python.org

Thank you…

- Please fill out the session survey in your mobile app
- Select Python An Introduction in the Mobile App
  - Use the Search Feature to quickly find this title
- Click “Technical Workshop Survey”
- Answer a few short questions and enter any comments



Understanding our world.