Python – An Introduction

Clinton Dow, Jibin Liu

Agenda

Python

- What is Python?
- Why Learn Python?
- Python Use Cases
- Intro to Python Syntax
- Writing and executing Python Scripts

ArcGIS

- ArcPy and Geoprocessing
- Writing Python scripts for Geoprocessing
- Python in Esri products

What is Python?

- **Python is an open-source programming language**
  - **Released in 1991 by Guido Van Rossum**
  - **Interpreted – no compilation**
  - **Interactive – REPL (Read, Evaluate, Print Loop)**
  - **Object-oriented**
- **Integrated into ArcGIS**
  - **Geoprocessing Scripts**
  - **Python Window**
  - **Field Calculator Expressions**
- **Desktop and Web GIS**
  - **arcpy**
  - **ArcGIS Python API**

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**

- **A "Batteries Included" Language**
- **Large Ecosystem of Open-Source Packages**
- **Great community (Conferences, User Groups, Online…)**

[1] https://www.python.org
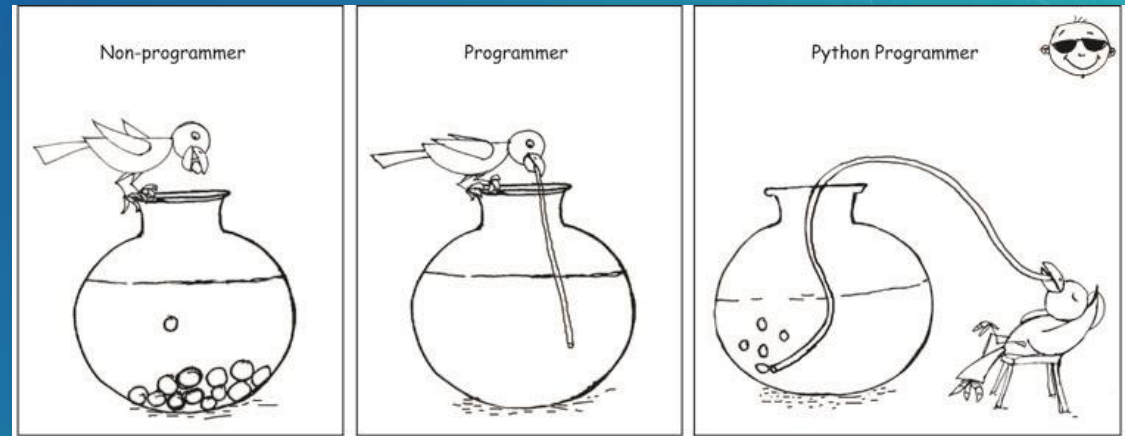
Why Learn Python?

Why Learn Python?

- **Accessible to new-comers**
    - **Top language for intro CS courses**
- **Large demand in multiple industries**
- **Create your own geoprocessing tools**
    - **Create suite of custom tools**
    - **Suit client's needs better than generic tools**
    - **Scheduling tasks**

- **A Versatile Language**
    - **"Glue" language that works with Operating System, Server and the Web**
- **Extend the capabilities of ArcGIS**
    - **Utilize third-party and/or open-source code in your scripts**
    - **Built-in package management**
- **Automate repetitive tasks**
    - **Saves time and money**
    - **Frees up analysts for non-trivial work**

Should I Learn Python 2 or Python 3?

- **ArcGIS Desktop**
  - **ArcGIS 10.5 - Python 2.7.12**
  - **Maintaining Existing Tools**
  - **Extending functionality of ArcMap, ArcCatalog**
  - **End of Official Support in 2020**

- **ArcGIS Pro**
  - **ArcGIS Pro 2.0 - Python 3.5.3**
  - **New functionality of Python and ArcGIS**
    - **Asynchrony**
    - **Reduced Memory Footprint**
  - **In active development**

Resources to Learn Python

- **Websites**
  - Python.org
    - Beginner's Guide
    - Language Reference
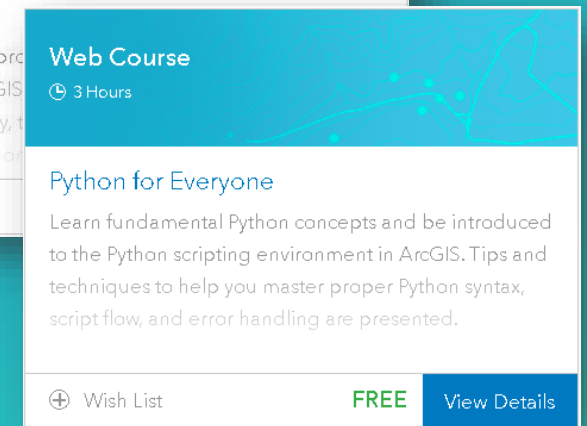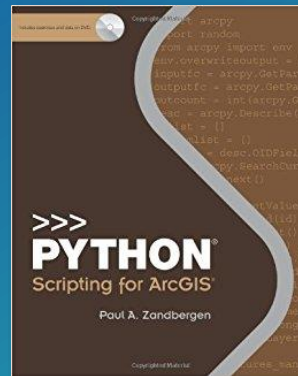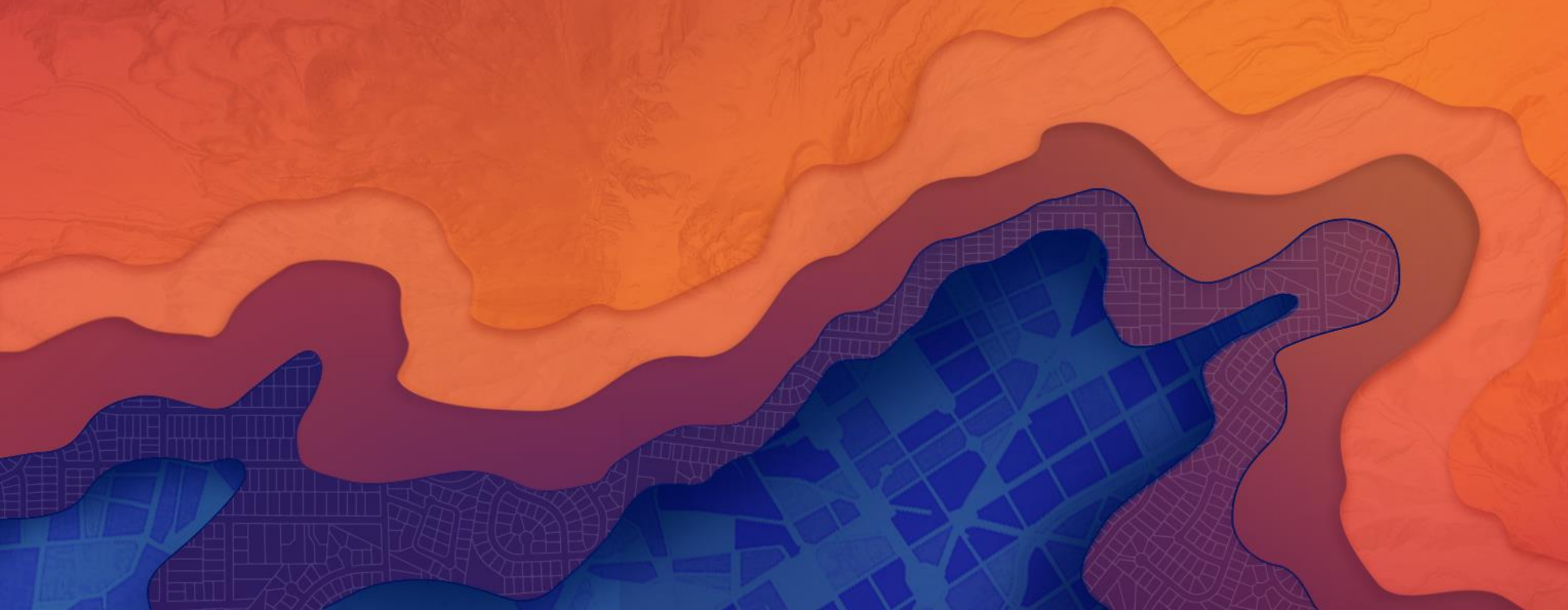 - Learnpython.org
  - Pluralsight.com
- **Esri Training**
  - Free and Paid Courses
- **Books**
  - Python Scripting for ArcGIS
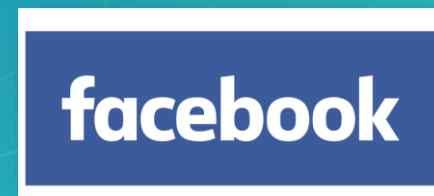 - Learning Python, 5th Edition

Where is Python Used?

- **Examples include:**
  - Web Applications
  - **Data Analysis**
  - **Entertainment**

- **Examples include:**
  - **Web Applications**
  - **Data Analysis**
  - **Entertainment**

Numerical Algorithms Group

Capital One

CONTINUUM ANALYTICS

Clover

elastic

GE

PIXAR
ANIMATION STUDIOS

MINECRAFT

- **Examples include:**
  - Web Applications
  - Data Analysis
  - Entertainment

Walt Disney
ANIMATION STUDIOS

EVE
ONLINE

DemonWare

Sid Meier's
CIVILIZATION IV

Python Syntax

Introduction to Python Syntax: Demo

- Math Calculations
- Variables
- Built-in Data Structures and their operations
- Conditions
- Loops
- Import libraries

Python Scripting

Executing Python Scripts

- **What is Python File (.py)?**
  - **A text file with .py extension, which python interpreter can read the instructions and execute them.**
  - **Recommended Syntax defined in Python Enhancement Proposal 8 (PEP8)**

- **Where to write the .py file?**
  - **Text editor (notepad), Python IDLE, other IDEs**

- **How to execute the .py file?**
  - **Double-click the .py file**
  - **Command line (python <path to .py file>)**
  - **IDEs**
  - **ArcGIS**
  - **System Service**
  - **Web Service**

Python Building Blocks

- *Module*: a Python file where functions live (.py)

- 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

Executing Python Scripts: Demo

- Double-click Python File
- From Command Line

Python in Esri

- **Python in Esri products**
  - arcgisscripting
  - ArcPy
  - Anaconda
  - ArcGIS Python API

2017

2015

2010

```
import arcpy
arcpy.Intersect_analysis()
```

2004

```
import arcgisscripting
gp = arcgisscripting.create()
```
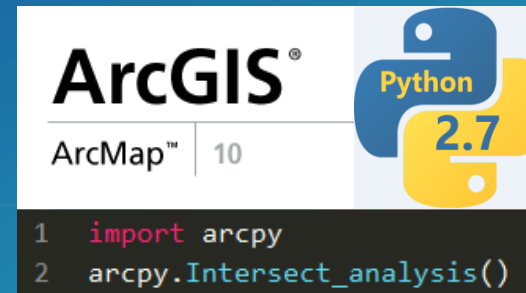
Python in Esri

- **Python in Esri products**
  - **arcgisscripting**
  - **ArcPy**
  - Anaconda
  - **ArcGIS Python API**

Conda embedded in Pro since 1.3

- "Conda is an open source package management system and environment management system for installing multiple versions of software packages and their dependencies and switching easily between them." -- *http://conda.pydata.org/docs/*

- **Conda solves limitations in core Python infrastructure**
  - **Handling dependencies**
  - **Locating, compiling Python libraries**
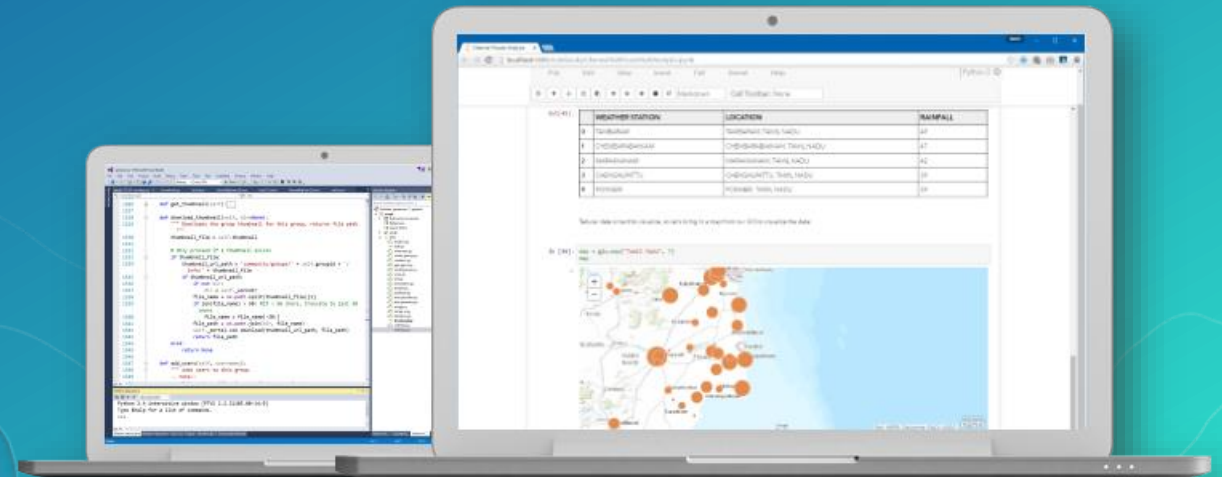 - **Managing multiple Python versions**

Python in Esri

• **Python in Esri products**

- **arcgisscripting**
- **ArcPy**
- **Anaconda**
- ArcGIS Python API

Script and automate your Web GIS

- A pythonic library to interoperate with Esri Web GIS Products

- Designed to integrate with the Jupyter Notebook, an increasingly popular tool for academics and data scientists.

Python in ArcGIS Desktop vs. ArcGIS Pro

Feature	ArcGIS 10.x	ArcGIS Pro
Python 2	√	
Python 3		√
Python Window	√	√
Python Script Tools	√	√
Python Add-Ins	√	
Conda Package Manager		√
Debugging Experience	√	Coming soon

ArcGIS Python window

- **Both Desktop and Pro have an embedded, interactive Python command line**

- **Access to Python and modules within ArcGIS applications**

- **Interact with maps and layers directly with Python code**
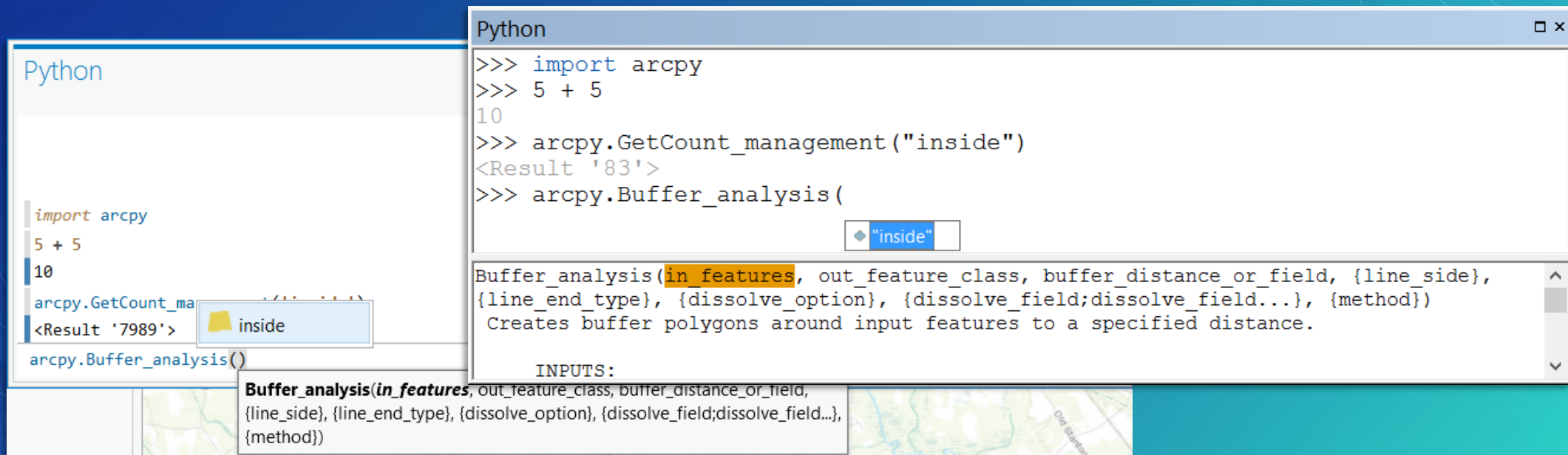
ArcPy

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

1. **Geoprocessing tools**
2. **Functions like ListFeatureClasses, Describe**
3. **Classes like Polygon, SpatialReference, FieldMap**
4. **Modules**
    a) Mapping: **arcpy**.*mapping* / **arcpy**.*mp*
    b) Data access: **arcpy**.*da*
    c) Map algebra: **arcpy**.*sa*
   d) Network Analyst: **arcpy**.*na*

ArcPy - functions

- An ArcPy function for many operations in ArcGIS UI

- Interact with ArcGIS Tool Dialogues

- Describe existing datasets

- Information about installation
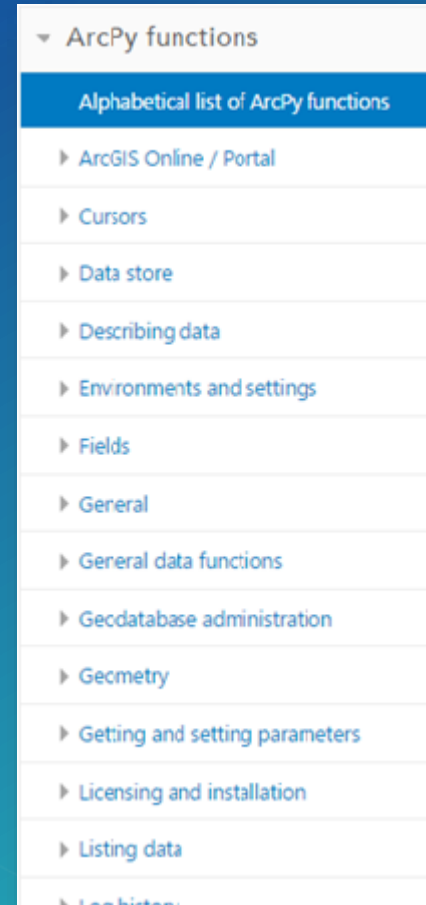
ArcPy - Classes

- **Python objects representing major base classes in ArcGIS.**

- **Extend ArcGIS objects for use with other systems.**

- **Customize behaviors of objects within your scripts.**
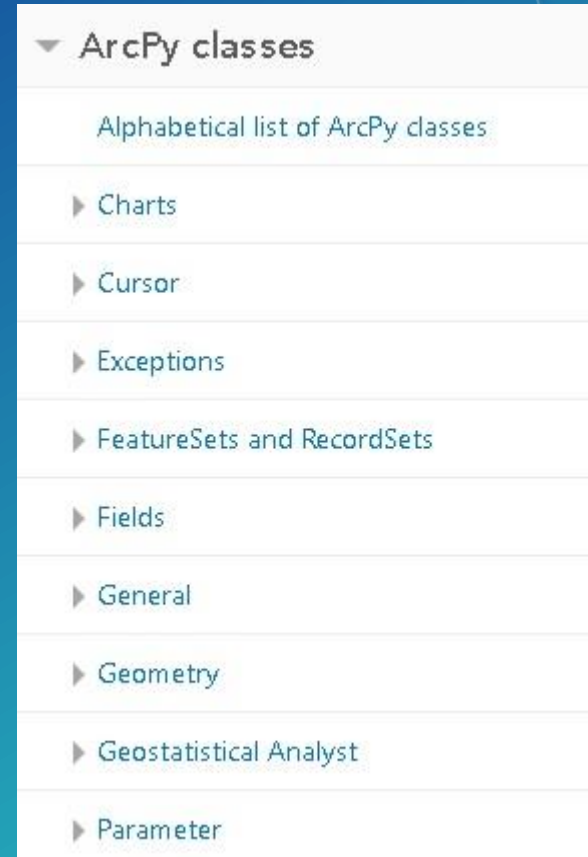
ArcPy - Geoprocessing environment settings

- **Control the processing environment of the tools you run**
  - **"Global" Environment Variables**
  - **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
```

ArcPy - 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*

▼ Listing data

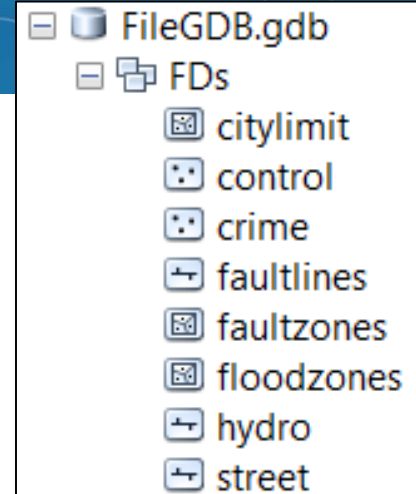| ListDatasets |
| ListFeatureClasses |
| ListFields |
| ListFiles |
| ListIndexes |
| ListRasters |
| ListTables |
| ListVersions |
| ListWorkspaces |

ArcPy - Batch processing (ListFeatureClasses)

```python
# 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)
```

FileGDB.gdb
  FDs
    citylimit
    control
    crime
    faultlines
    faultzones
    floodzones
    hydro
    street

ArcPy - Getting data properties

- **Describe functions reads data properties**
  - Like the properties window when right-click on the data

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

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

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

ArcGIS Python Window and ArcPy: Demo

- Open and execute python commands in ArcGIS Python Window
- ArcPy : environment setting
- ArcPy : batch processing
- ArcPy : getting data properties

Run geoprocessing tools

- **import arcpy**

- **Follow tool syntax**
  - **arcpy.toolname_toolboxalias(parameters)**
    or
    arcpy.toolboxalias.toolname(parameters)

- **How do I use a specific tool?**
  - **Tool help page**
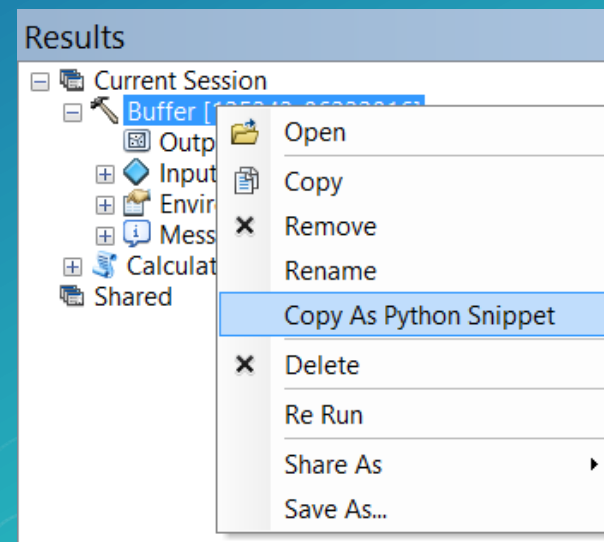 - **Copy as Python Snippet**
  - **help(arcpy.Buffer_analysis)**

Syntax
Buffer_analysis (in_features, out_feature_class,
buffer_distance_or_field, {line_side}, {line_end_type},
{dissolve_option}, {dissolve_field}, {method})

Code Sample

Buffer example 1 (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/majorrdsBuffered", "100 Feet", "FULL",
```


Results
Current Session
  Buffer [   ]
    Outp   Open
    Input   Copy
    Envir   Remove
    Mess   Rename
  Calculat   Copy As Python Snippet
Shared   Delete
                Re Run
                Share As
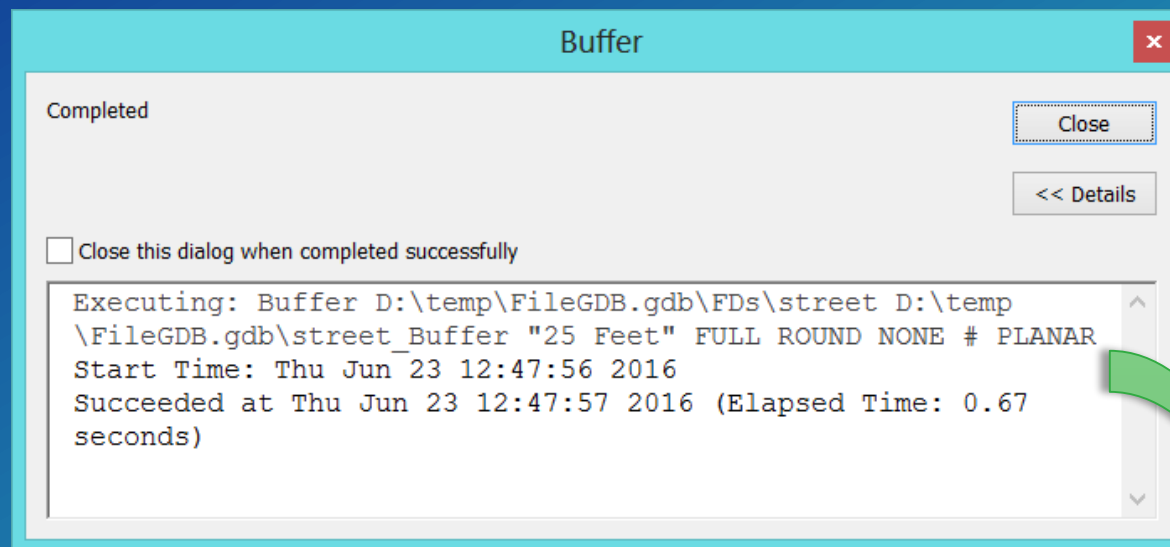   Save As...

Writing a Python script to chain geoprocessing tools: Demo

- Build Python script for geoprocessing tools

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()**
  - **arcpy.AddMessage()**
  - **arcpy.AddWarning()**
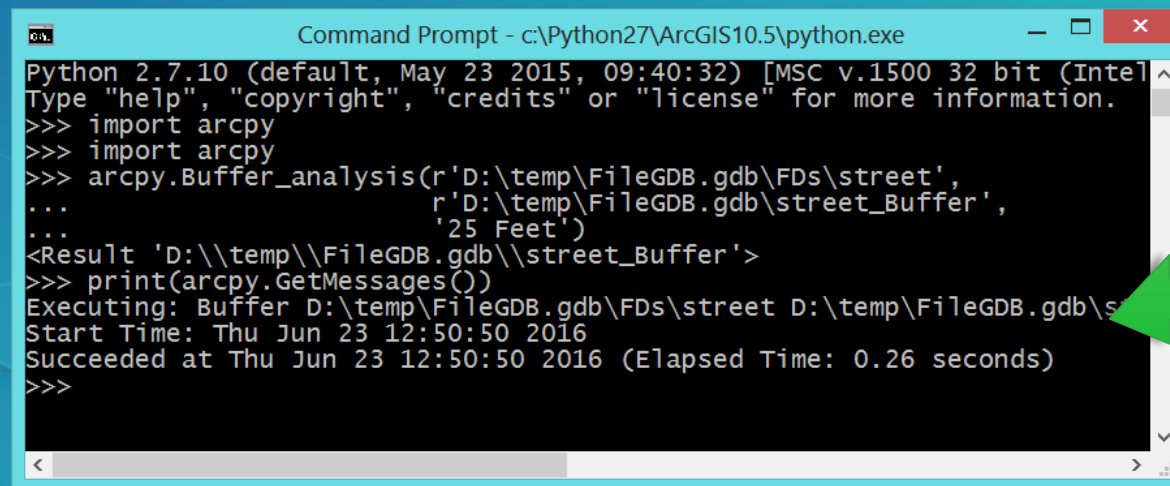
Troubleshooting

- **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**
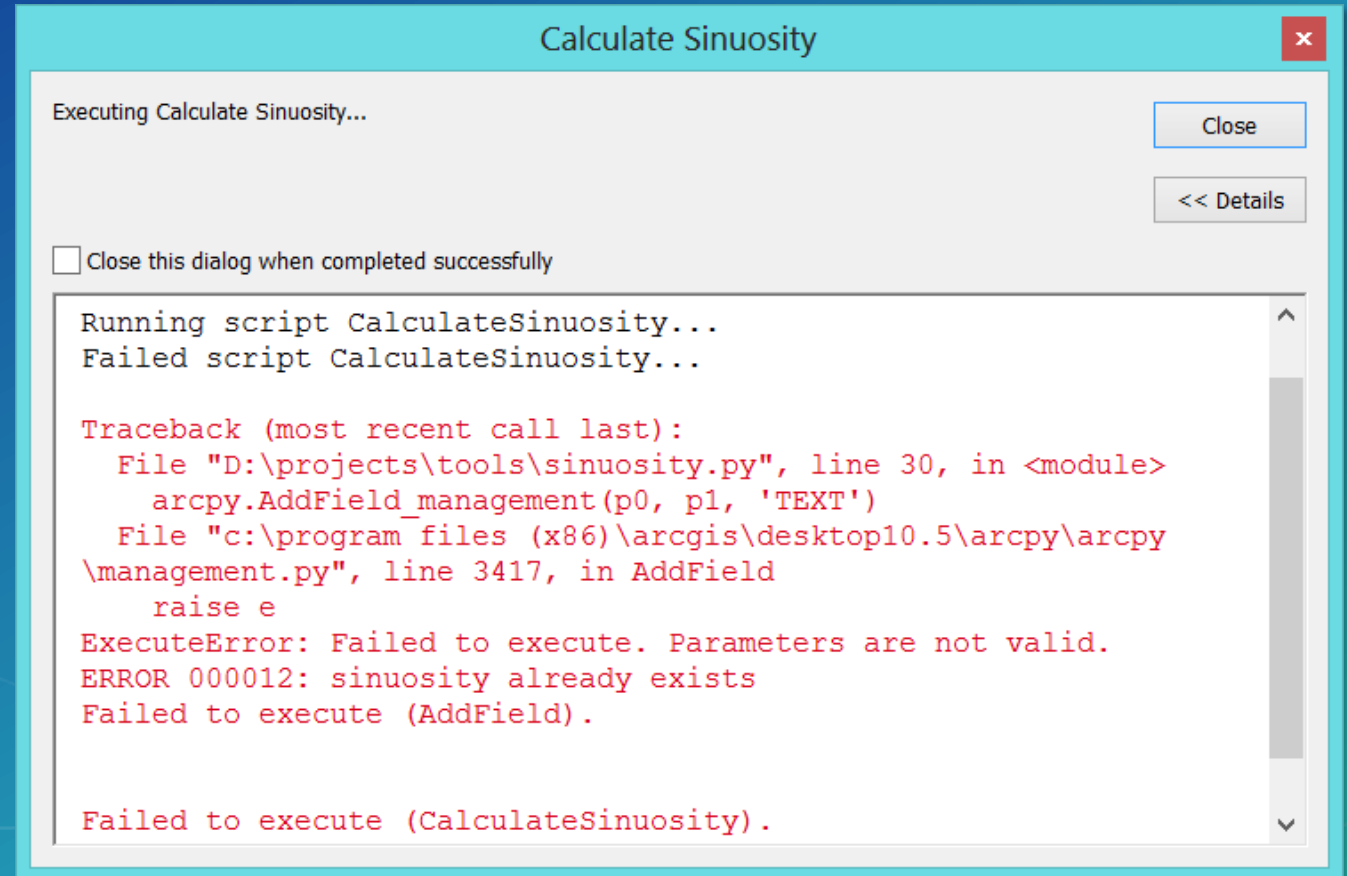
Further Learning at Esri UC 2017:

	Tues	Wed	Thurs	Fri
Python: An Introduction	8:30am		1:30pm	
Python: Beyond the Basics	10:15am		3:15pm	
Python: Building Geoprocessing tools	1:30pm			9:00am
ArcGIS: Integrating Python and Conda	1:00pm			
Python: Exploring the Ecosystem			12:30pm	
Python: ArcPy Tips and Tricks		1:30pm		
Working with Python in ArcGIS Pro			10:00am	
Python - Tips and Tricks for Working with Cursors		4:00pm		
Troubleshooting Python Issues in ArcGIS Desktop	1:30pm			
Network Analysis using Python		3:30pm		
Python: Raster Analysis	8:30am			
Building Python Raster Functions	10:30am			
Getting Started With Map Algebra Using the Raster Calculator and Python			9:30am	
Python: Introduction to Map Automation		1:30pm		
Python: Map Automation in ArcGIS Pro		3:15pm		
Python - Tips and Tricks for Working with Arcpy Mapping			12:00pm	

Please Take Our Survey on the **Esri Events App!**

Download the Esri Events app and find your event

Select the session you attended

Scroll down to find the survey

Complete Answers and Select "Submit"

THANK YOU!

esri | THE SCIENCE OF WHERE