esri Indonesia

Beginner Guide: Quick Tour ArcPy

Citra Hasana Sagala
Solution Engineer

Agenda

Python

- What is Python?
- Why Learn Python?
- Python Use Cases

ArcGIS

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

What is Python?

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

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 10.7.1 - Python 2.7.16
  - Maintaining Existing Tools
  - Extending functionality of ArcMap, ArcCatalog
  - End of Official Support in 2020

- ArcGIS Pro
  - ArcGIS Pro 2.4 - Python 3.6.8
  - New functionality of Python and ArcGIS
    - Deep Learning Tools
    - Parallel processing
  - In active development

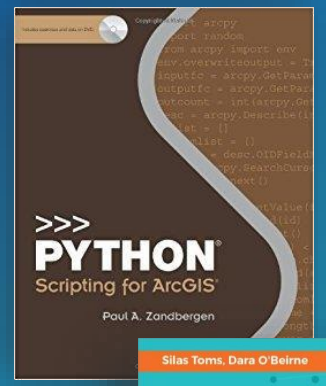
Resources to Learn Python

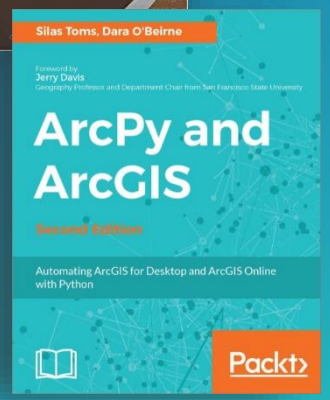
- **Websites**
  - Python.org
    - Beginner's Guide
    - Language Reference
  - Learnpython.org
  - Online Course
- **Esri Training**
  - Free and Paid Courses
- **Books**
  - Python Scripting for ArcGIS
  - Learning Python, 5th Edition

learnpython.org

UDACITY

DataCamp coursera





Where is Python Used?

Mainly used for

Web and Internet Development
Desktop GUI Apps
Scientific and Numeric Apps
Software Development
Database Access
Enterprise Apps
Robotics
Computer Vision
Machine Learning
Data Analysis
Scripting
Network

ArcPy and Geoprocessing

Python in Esri

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

2004

ArcGIS 9
ArcMap

```
1  import arcgisscripting
2  gp = arcgisscripting.create()
```

2010

ArcGIS
ArcMap™ 10
Python 2.7

```
1  import arcpy
2  arcpy.Intersect_analysis()
```

2015

ArcGIS® Pro
python
3 Python

2017

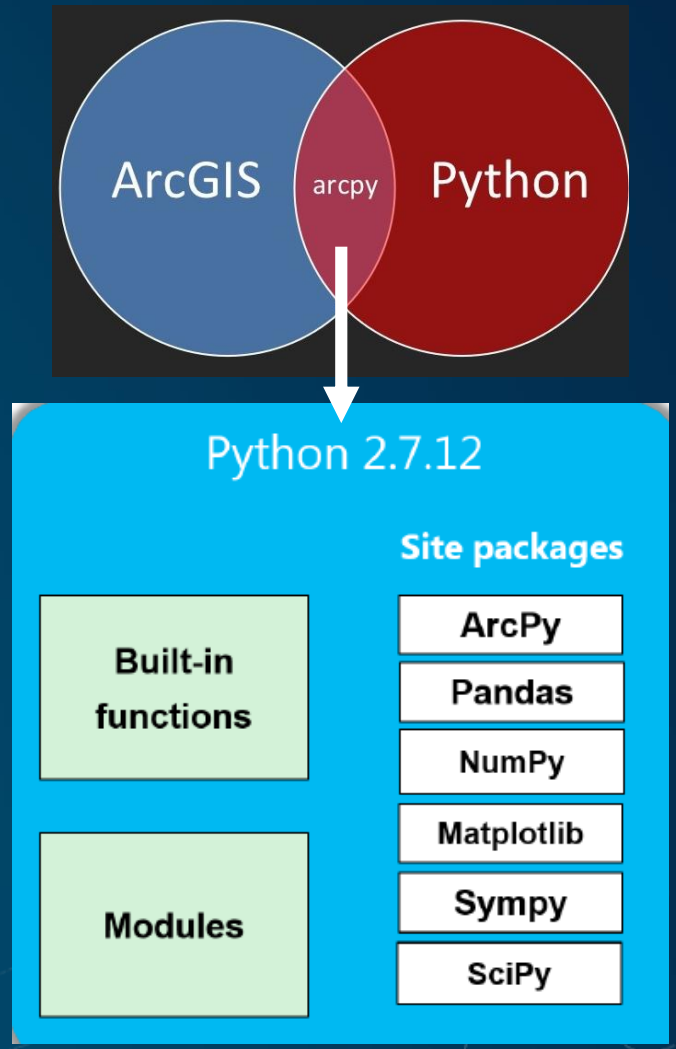
ANACONDA
jupyter

2019

ArcGIS Notebook Server
Spatial analysis meets data science

Python in Esri

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



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/*

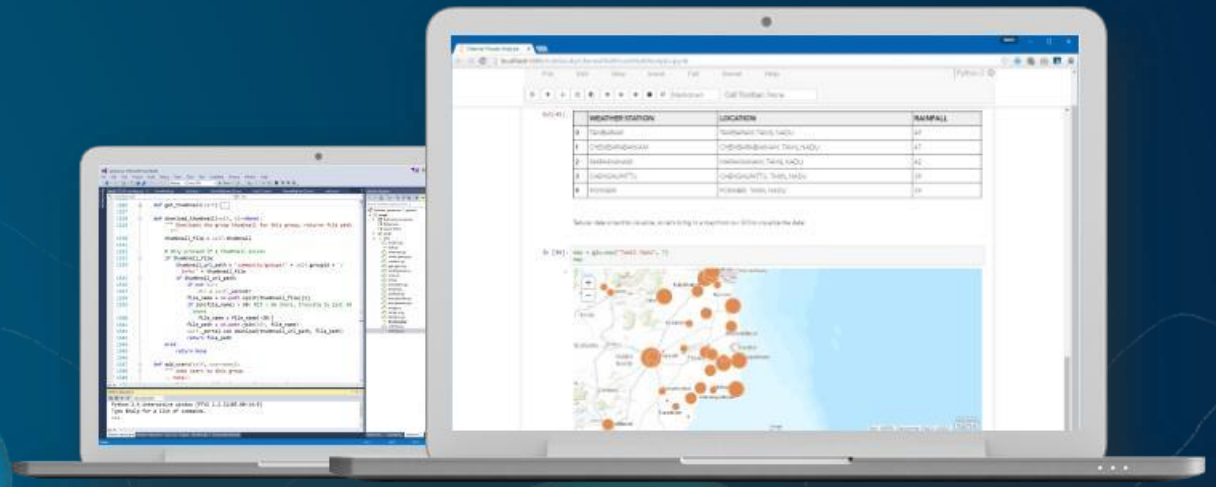
- **Conda solves limitations in core Python infrastructure**
  - **Handling dependencies**
  - **Locating, compiling Python libraries**
  - **Managing multiple Python versions**

Python in Esri

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

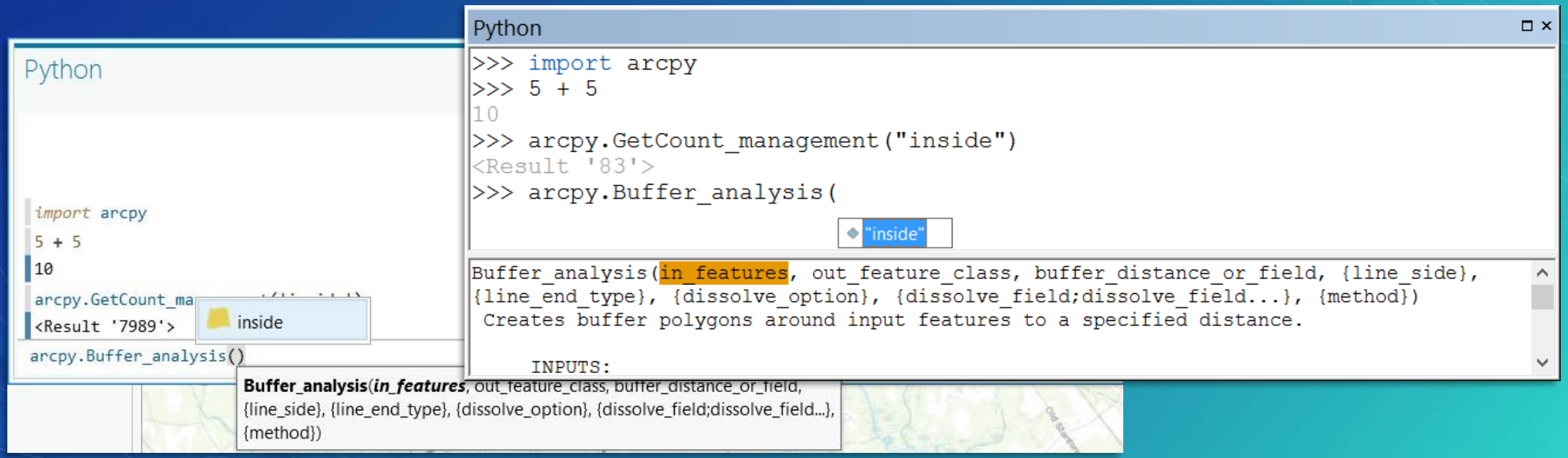
Script and automate your Web GIS

- 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.



ArcGIS Python window

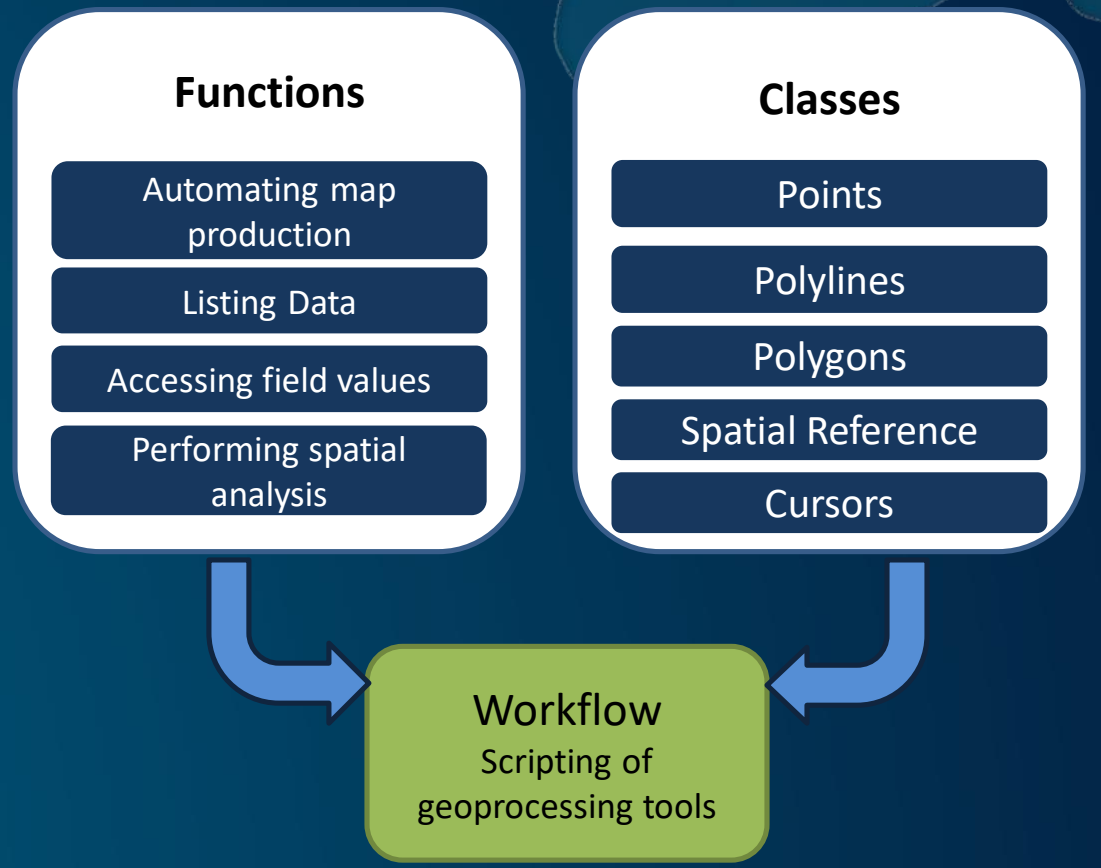
- Both Desktop and Pro have an embedded, interactive Python command line
- Access to Python and modules within ArcGIS applications
- Interact with maps and layers directly with Python code



ArcPy

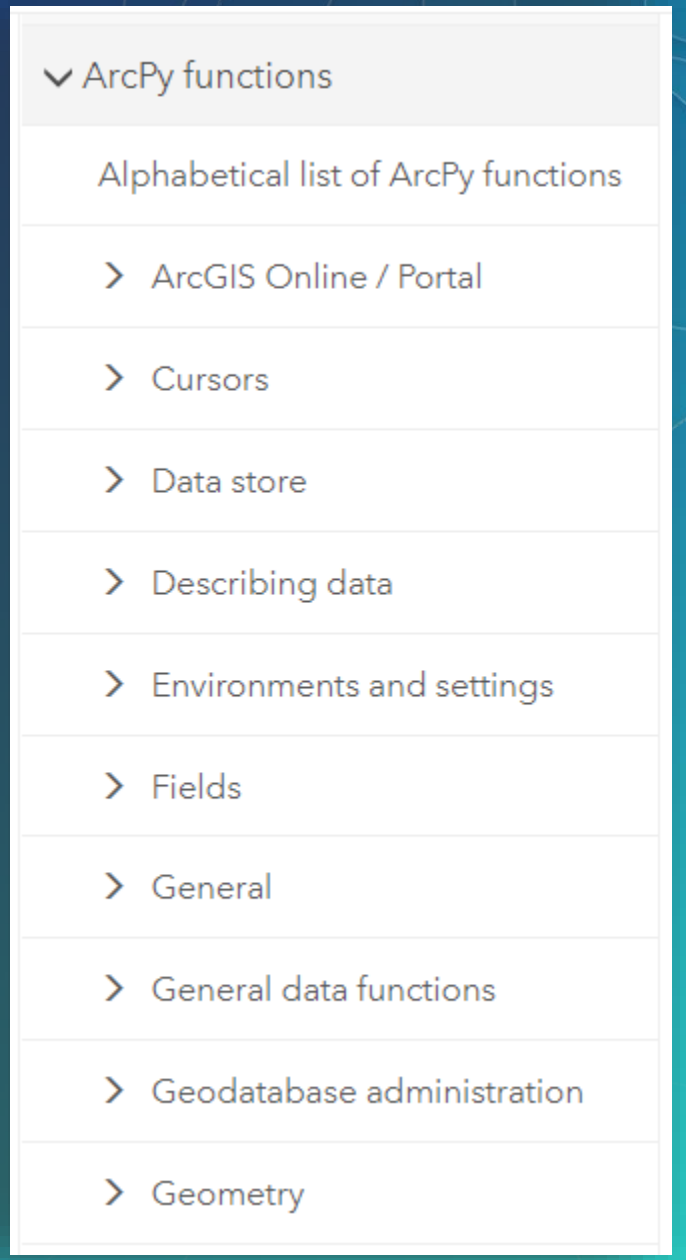
- Access point to ArcGIS functionality through Python
  - Desktop, Server, Engine, and Pro

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*



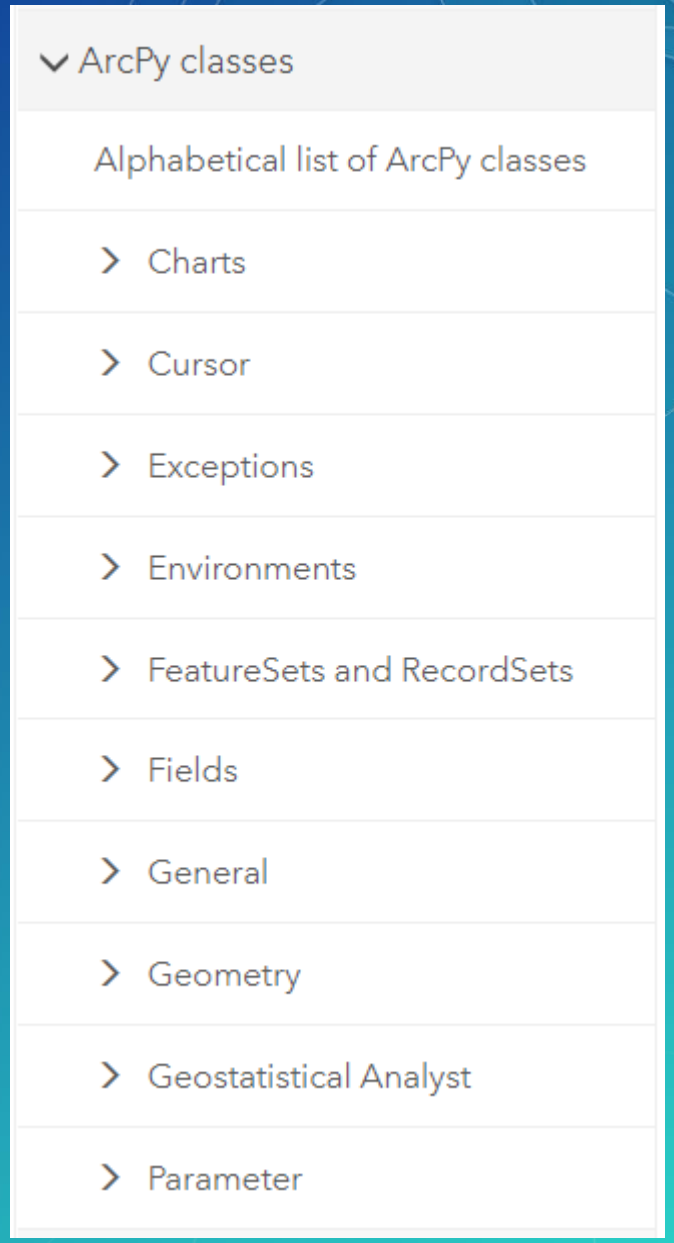
ArcPy - functions

- 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

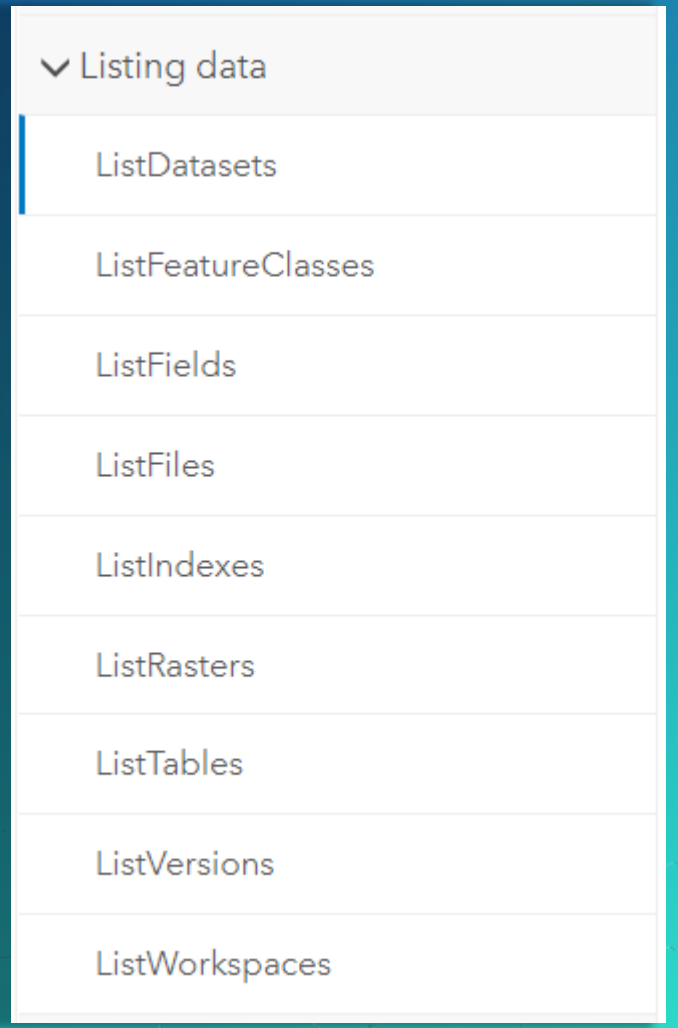
- Control the processing environment of the tools you run
  - “Global” Environment Variables
  - See tool help for honored environments
- Productivity and code cleanup
- Environments are properties on arcpy.env (over 50)

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

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

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

ArcPy - Batch processing

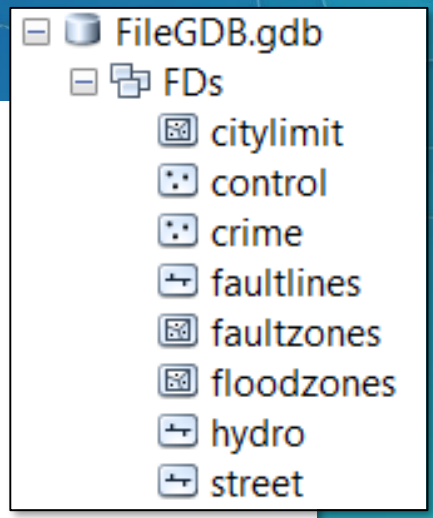
- **Automating a process to run multiple times**
  - **Clip every feature class in a geodatabase to a common boundary**
  - **Calculate statistics for every raster in a folder**

- **List functions used in Python to perform batch processing**
  - ***Also arcpy.da.Walk***



ArcPy - 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)
```



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

```
# 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

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

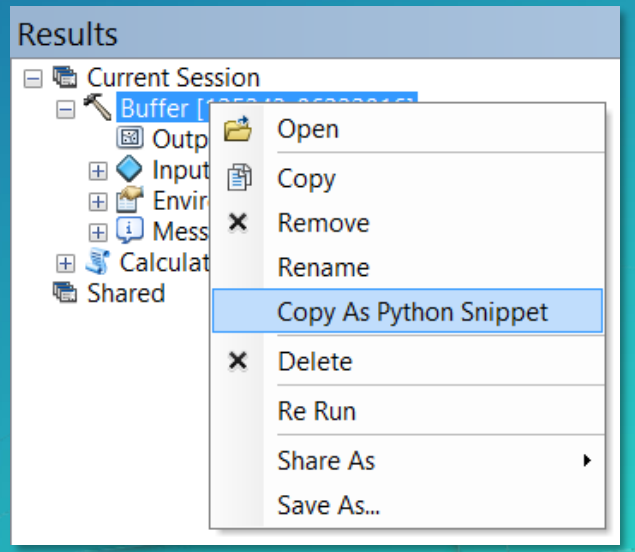
```
Buffer_analysis (in_features, out_feature_class,
buffer_distance_or_field, {line_side}, {line_end_type},
{dissolve_option}, {dissolve_field}, {method})
```

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",
```

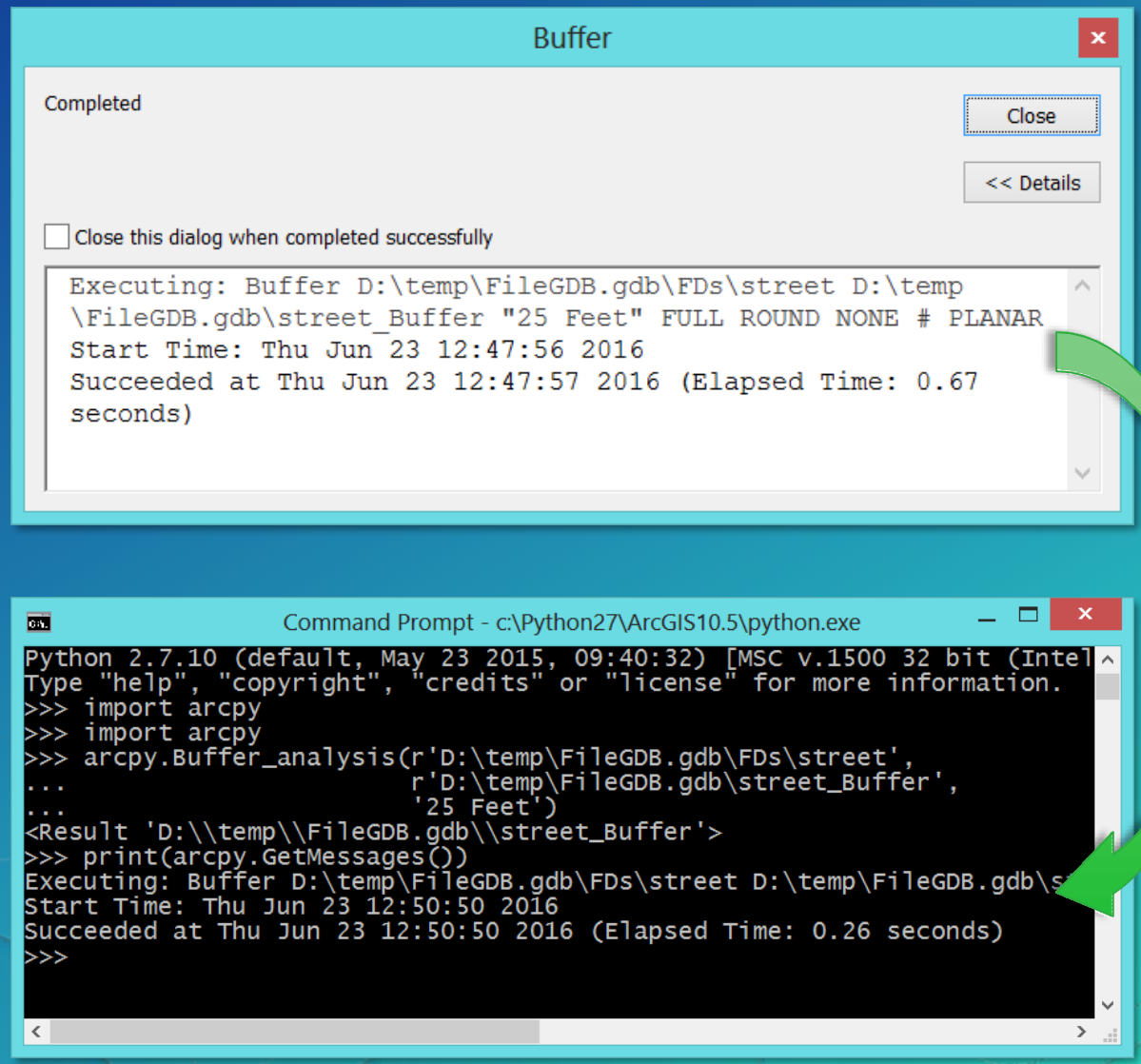


Writing a Python script to make geoprocessing tools: Demo

- Build Python script for geoprocessing tools

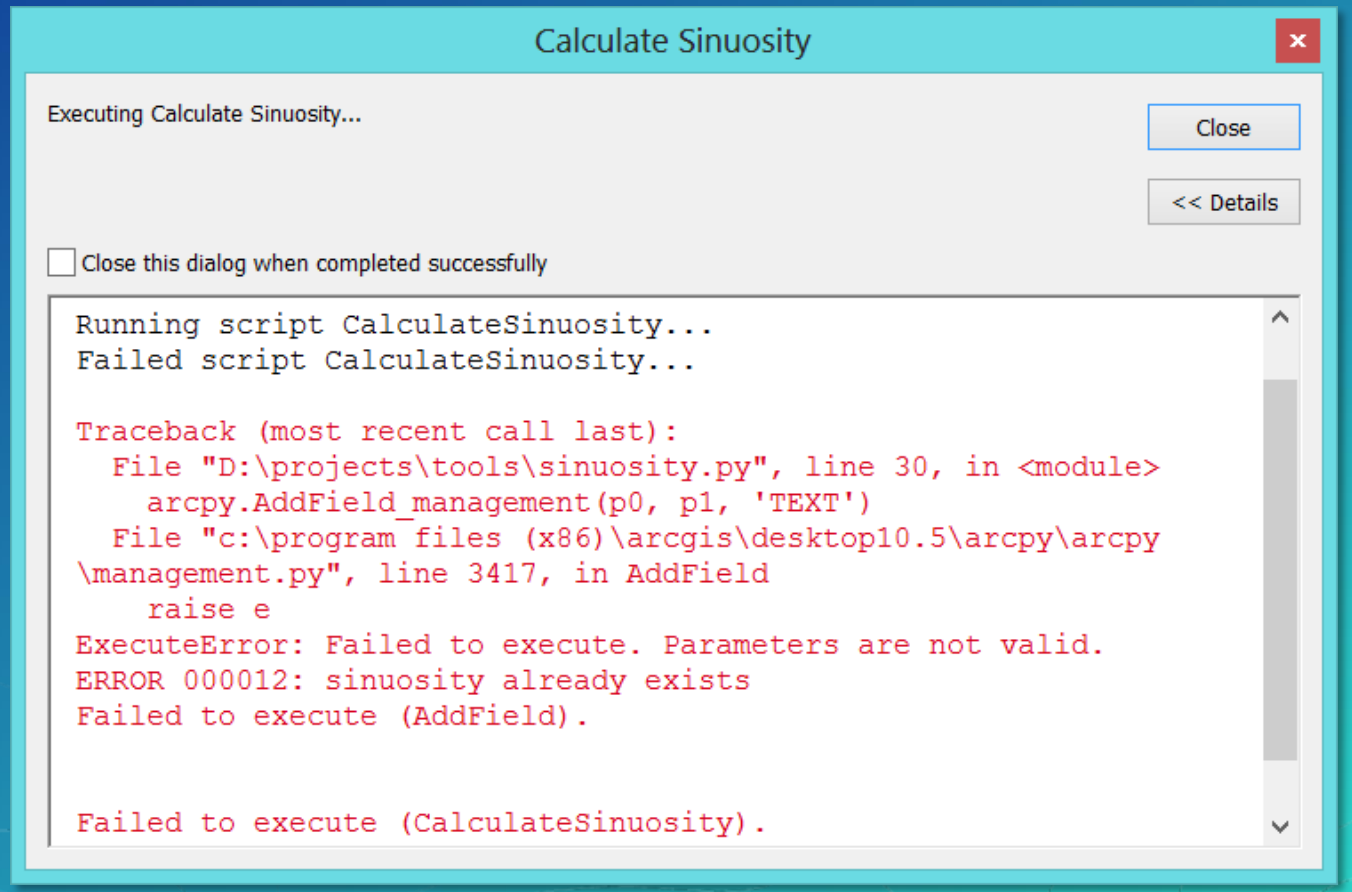
Geoprocessing tool messages

- 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



Questions?

Terima Kasih

esri Indonesia