ArcPy: An Introduction

Danny McVey

Hannes Ziegler

2021 ESRI USER CONFERENCE

Agenda

- About the Python language
- About ArcPy
  - List functions
  - Describe function
- Additional topics

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
- Perform (repeatable) analysis , data processing, administration…
- Extend ArcGIS Pro and Desktop
  - geoprocessing tools
  - raster functions
- Extend ArcGIS services with geoprocessing tasks

- Long and continued commitment
  - since 2004
  - actively part of future plans

Python Everywhere

Language Basics

Python Language Basics

- Dynamically Typed
- Common Types Supported
  - Strings
  - Numbers
  - Functions

```
a = 1
b = 2
c = a+b

print(c)
```

```
fc = r"C:\UC2019\Demos\Demos.gdb\MajorAttractions"
```

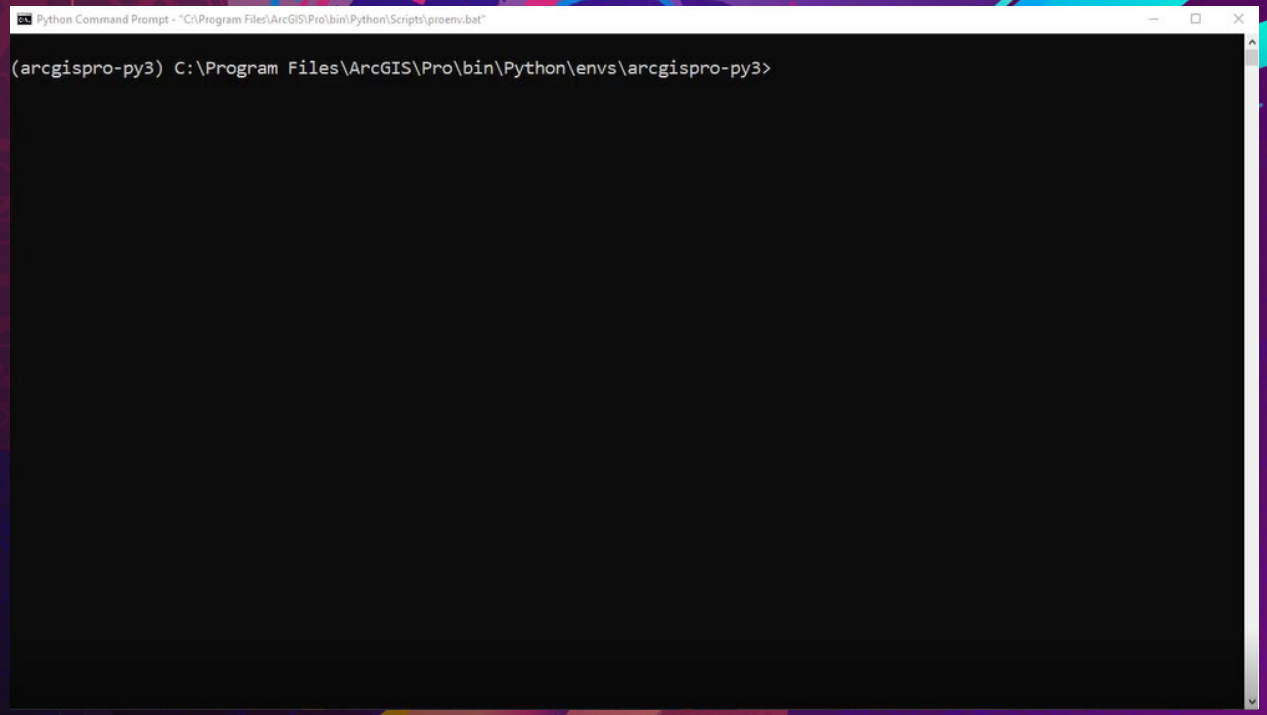
Python Language Basics

- Files
- Lists
- Dictionaries

```
# Xml processing
opnXml = open(r"C:\UC2019\Demos\attractions.xml",'r')
```

```
#            0       1       2
flds = ["NAME","ADDR","EMP"]
```

```
web_map_properties = {'title':mpTitle,
                      'type':'Web Map',
                      'snippet':mpSnippet,
                      'tags':'PM 2.5','text':1}
```



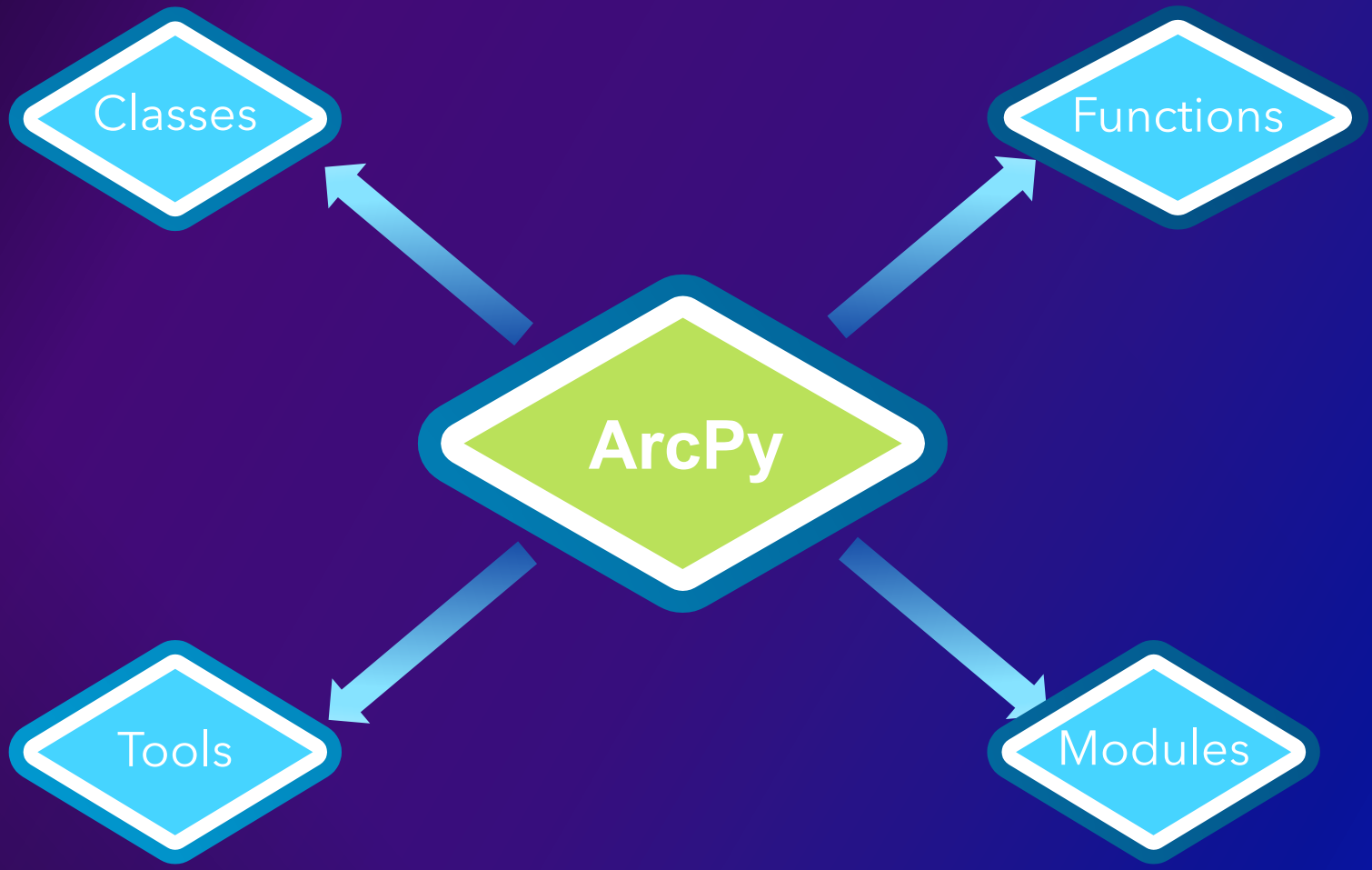
Demo: Python & ArcPy

ArcPy

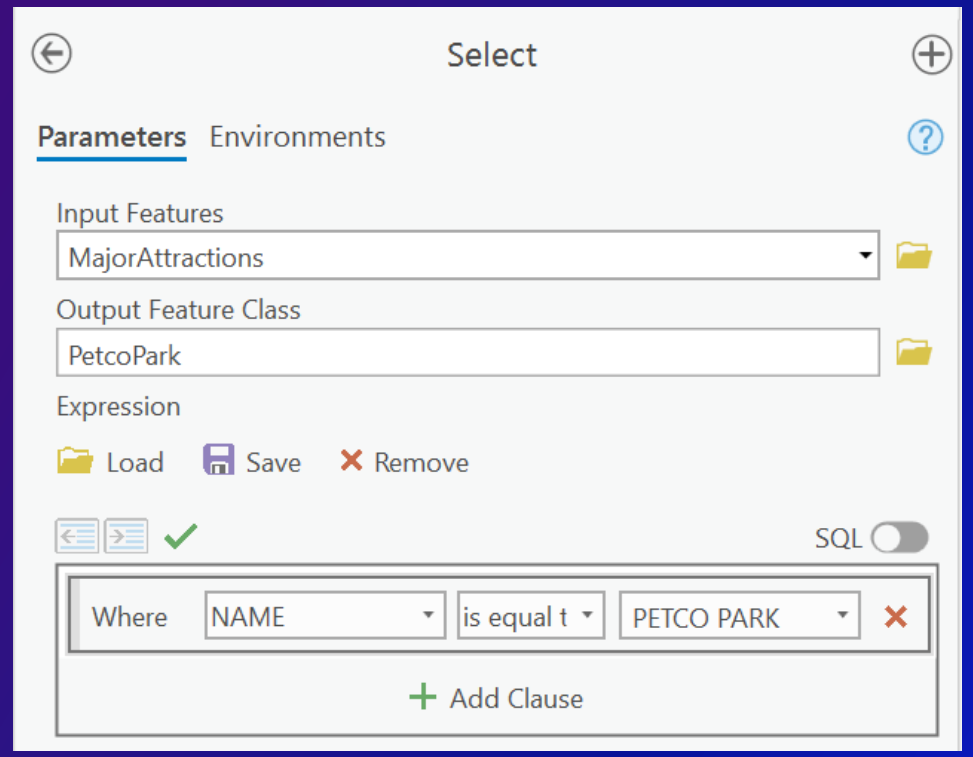
what is ArcPy?

- a python site-package
- a python api for ArcGIS Pro, ArcGIS Server, ArcGIS Desktop

```
print(9*9)
81

import arcpy
arcpy.Buffer_analysis(...)
```


Classes
Functions
ArcPy
Tools
Modules

ArcPy Tools




```
arcpy.Select_analysis("MajorAttractions","PetcoPark", "Name = 'PETCO PARK'")
```

Accessing Tools and Syntax help

Select (Analysis Tools)

Extracts features from an input feature class or input feature layer, typically using a select or Structured Query Language (SQL) expression, and stores them in an output feature class.

Syntax

```
Select(in_features, out_feature_class, {where_clause})
```

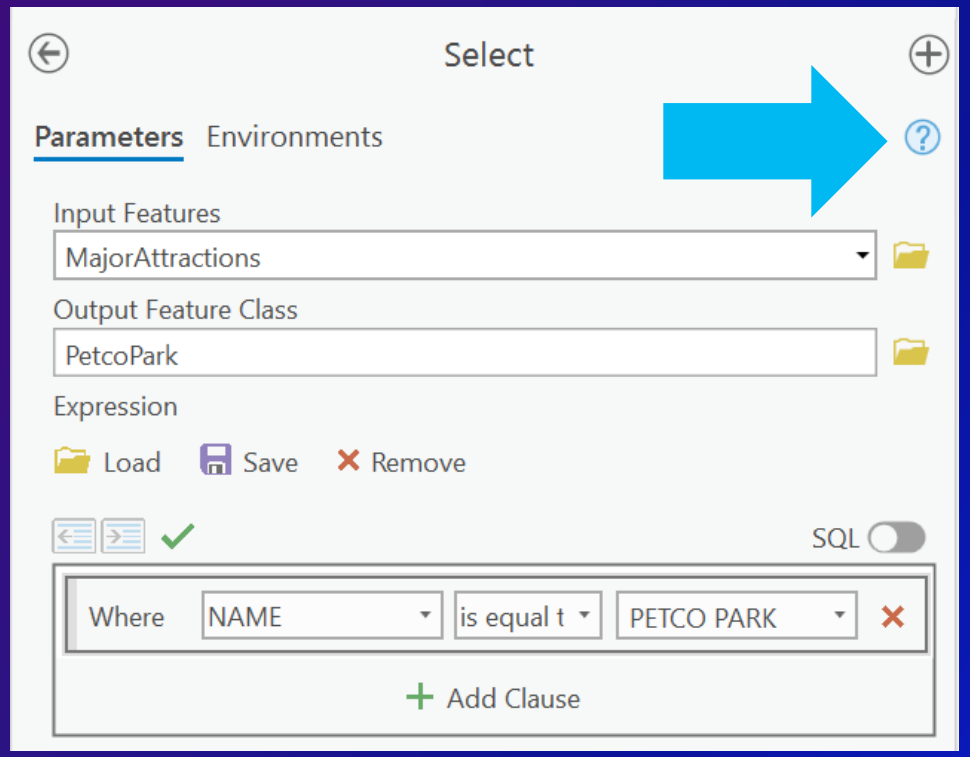
Parameter	Explanation	Data Type
in_features	The input feature class or layer from which features are selected.	Feature Layer
out_feature_class	The output feature class to be created. If no expression is used, it contains all input features.	Feature Class
where_clause (Optional)	An SQL expression used to select a subset of features. For more information on SQL syntax, see SQL reference for query expressions used in ArcGIS.	SQL Expression

Code sample

Select example 1 (Python window)

The following Python window script demonstrates how to use the `Select` function in immediate mode.

```
import arcpy
arcpy.env.workspace = "C:/data"
arcpy.Select_analysis("majorrds.shp", "C:/output/majorrdsClass4.shp", '"CLASS" = \'4\'')
```



ArcPy Functions

- General data functions
  - CreateScratchName
  - CreateUniqueName
  - Exists
  - ParseTableName
  - TestSchemaLock
  - ValidateTableName

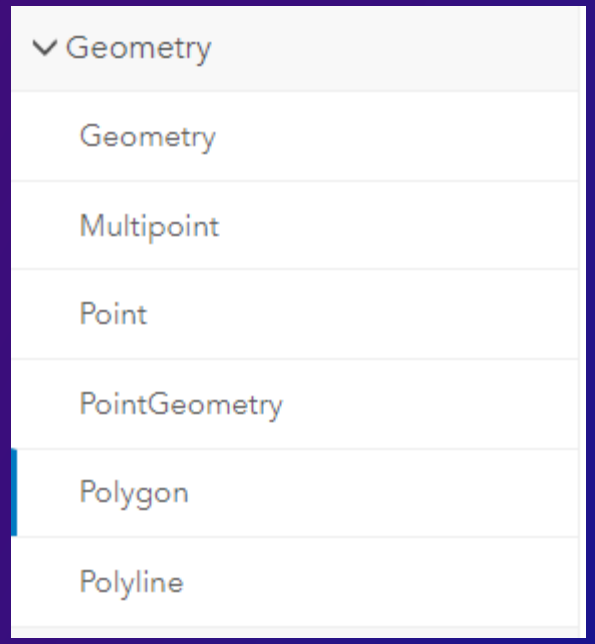
- Geodatabase administration
  - AcceptConnections
  - DisconnectUser
  - ListUsers

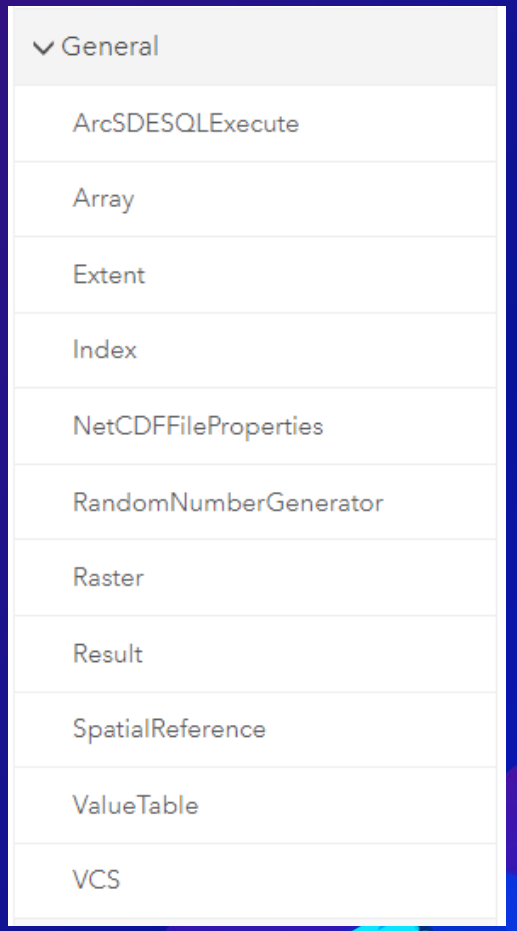
- Geometry
  - AsShape
  - FromGeohash
  - FromWKB
  - FromWKT

```
arcpy.Exists('MajorAttractions')
```

ArcPy Classes

- Used to help code
- Create geometry
  - Points
  - Polygons
  - Spatial Reference
- Provide field information

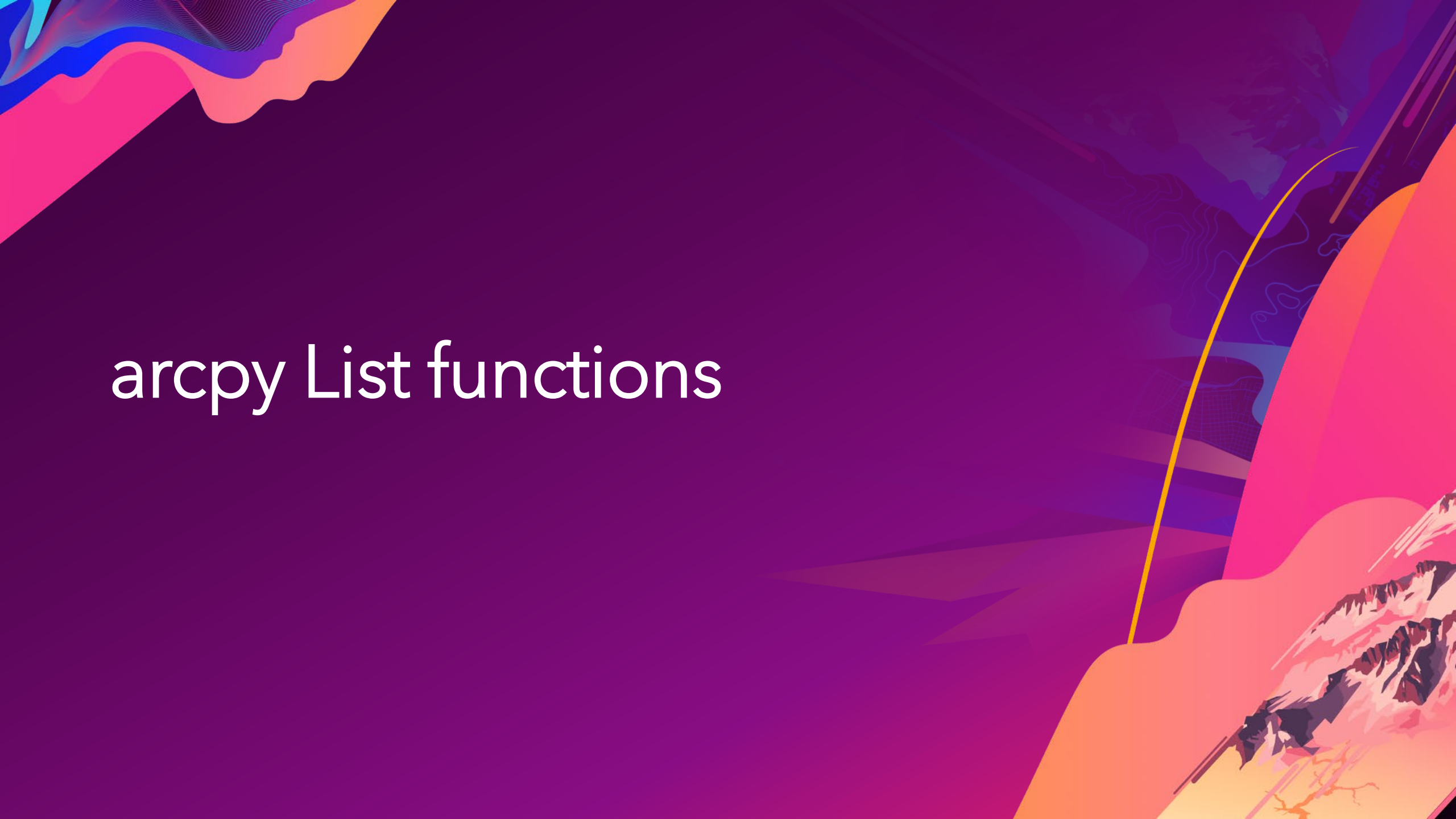




ArcPy Modules

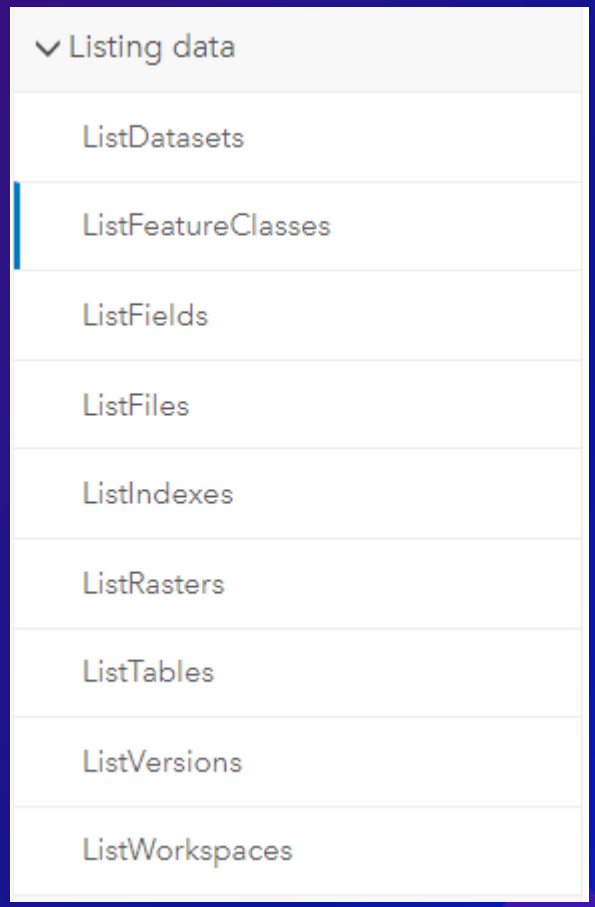
- collection of related functionality
  - arcpy.da
  - arcpy.metadata
  - arcpy.nax

arcpy List functions



Listing Functions

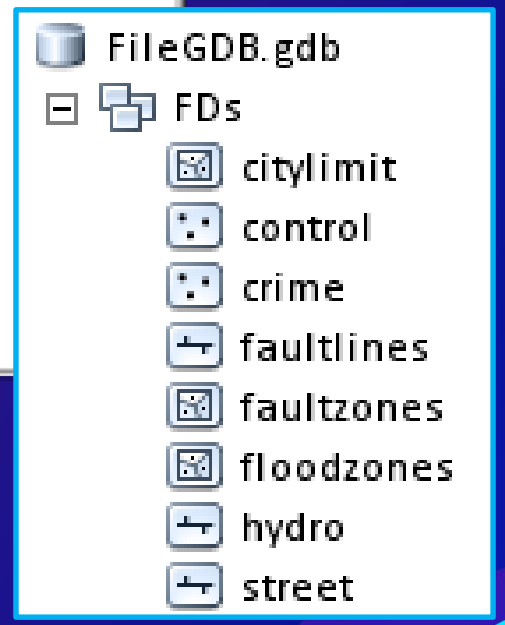
- **Return Python list**
  - **String**
  - **Object**
- **Minimal Parameters needed**



arcpy.ListFeatureClasses

- ListFeatureClasses ({wild_card}, {feature_type}, {feature_dataset})

```
# Set the workspace
arcpy.env.workspace = "C:/Data/FileGDB.gdb/FDs"

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

# Print the names of the feature classes
for fc in fcList:
    print(fc)
```



arcpy.da.Walk

- modeled after python's `os.walk`

```
ws = r"c:\projects\project_x"
walk = arcpy.da.Walk(ws, datatype="FeatureClass", type="Polygon")

for dirpath, dirnames, filenames in walk:
    for filename in filenames:
        feature_classes.append(os.path.join(dirpath, filename))
```

Python List Functions

- index
  - where in the list is an item
- sort
  - A to Z a to z
- count
  - how many times an item occurs in the list

```
my_list = arcpy.ListFeatureClasses()

len(my_list)
9

my_list.index('counties')
4

my_list = sort(my_list)
```

arcpy.Describe

Describe Function

Describe function reads data properties

Returns an object with properties

- Data type
- Shape type
- Spatial reference
- Fields

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

print(desc.shapeType)
Polyline
```

Describe Function

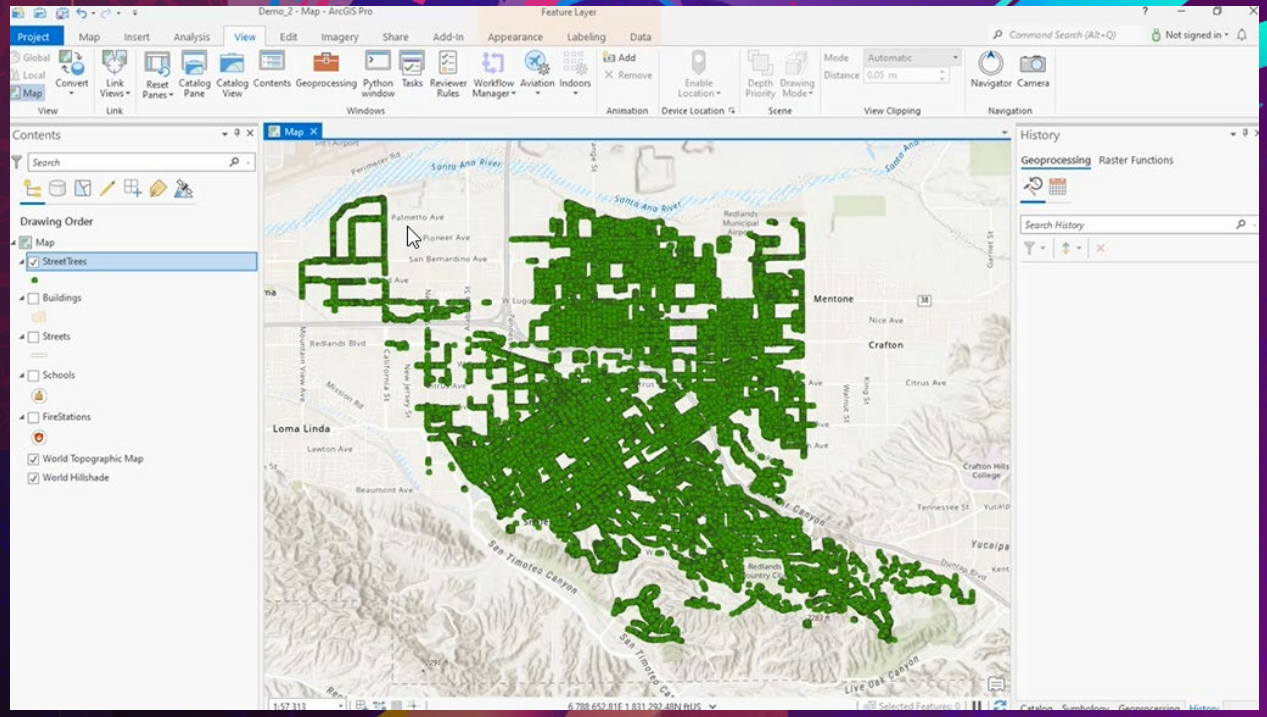
- Programmatically understand your data
- Useful when creating tools
  - Geometry Type
  - Fields
  - SpatialReference
  - RasterBands

```
desc = arcpy.Describe("C:/demos/MajorAttractions")
if desc.shapeType == "Point":
    ...

desc = arcpy.da.Describe("C:/demos/MajorAttractions")
print(desc)
{'OIDFieldName': 'OBJECTID',
 'dataType': 'FeatureClass',
 'shapeType': 'Point',
  ...
}
```

da.Describe Function

- Returns Python Dictionary
- Same Describe properties

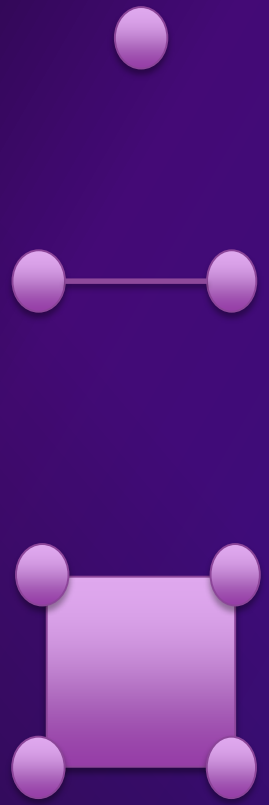
```
desc = arcpy.da.Describe("C:/demos/MajorAttractions")

print(desc)
{'OIDFieldName': 'OBJECTID',
 'dataType': 'FeatureClass',
 'shapeType': 'Point',
 'spatialReference': <SpatialReference object at 0x19>,

  ...
}
```



Demo: Python in Pro, List & Describe

Additional Topics

Creating Geometry



```
pnt = arcpy.Point(-22.570,63.968)
```

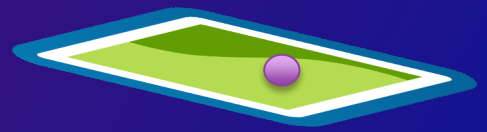
```
line = arcpy.Polyline(
arcpy.Array([arcpy.Point(-22.570,63.968),
             arcpy.Point(-22.573,63.969)
]))
```

```
Polygon = arcpy.Polygon(
arcpy.Array([arcpy.Point(-22.570,63.968),
             arcpy.Point(-22.573,63.969),
             arcpy.Point(-22.575,63.968)
])
)
```

Geometry Operators

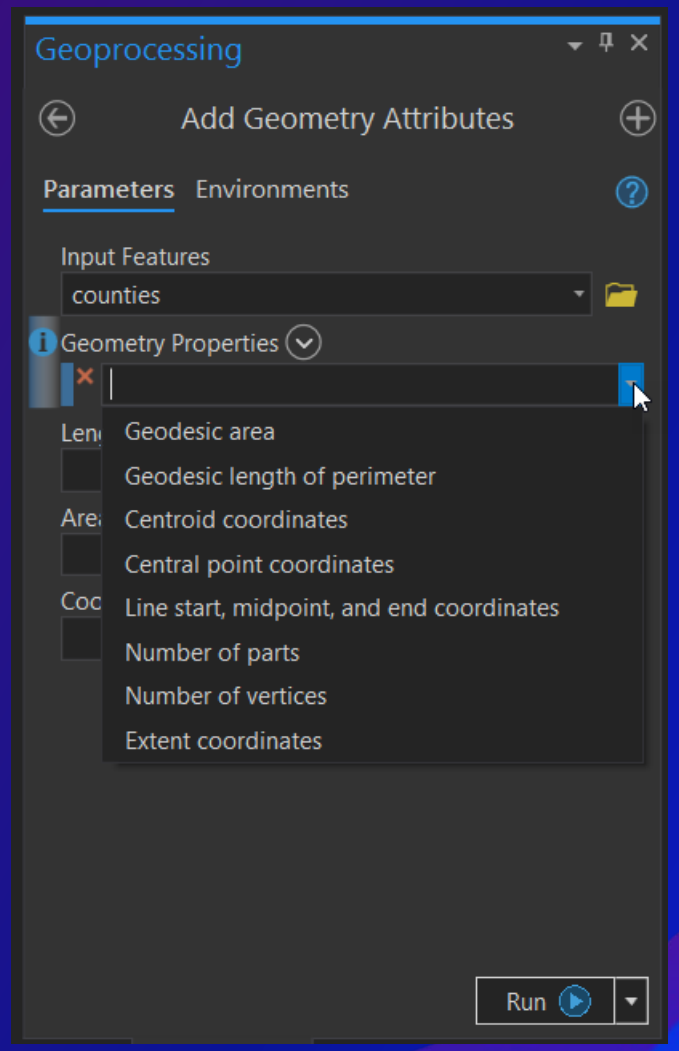
- Return Values
  - Boolean
  - Geometry/s
- Relational:
  - Is a point within a polygon
  - Return Boolean Values

```
sr = arcpy.SpatialReference(4326)
pt = arcpy.Point([-114.1, 51.0], sr)

print(pt.within(some_polygon))
True

print(pt.buffer(5))
<Polygon object at 0x1>
```



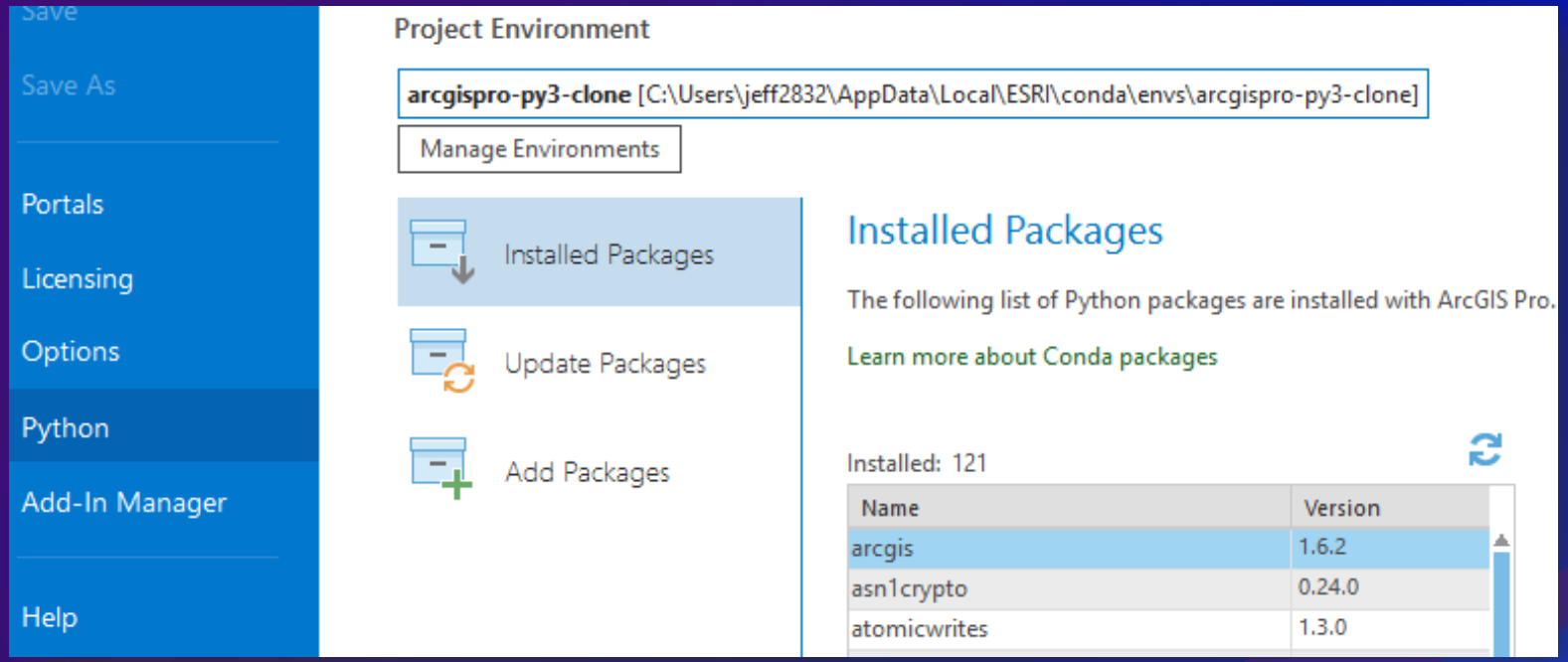
Extending ArcGIS Pro with gp tools

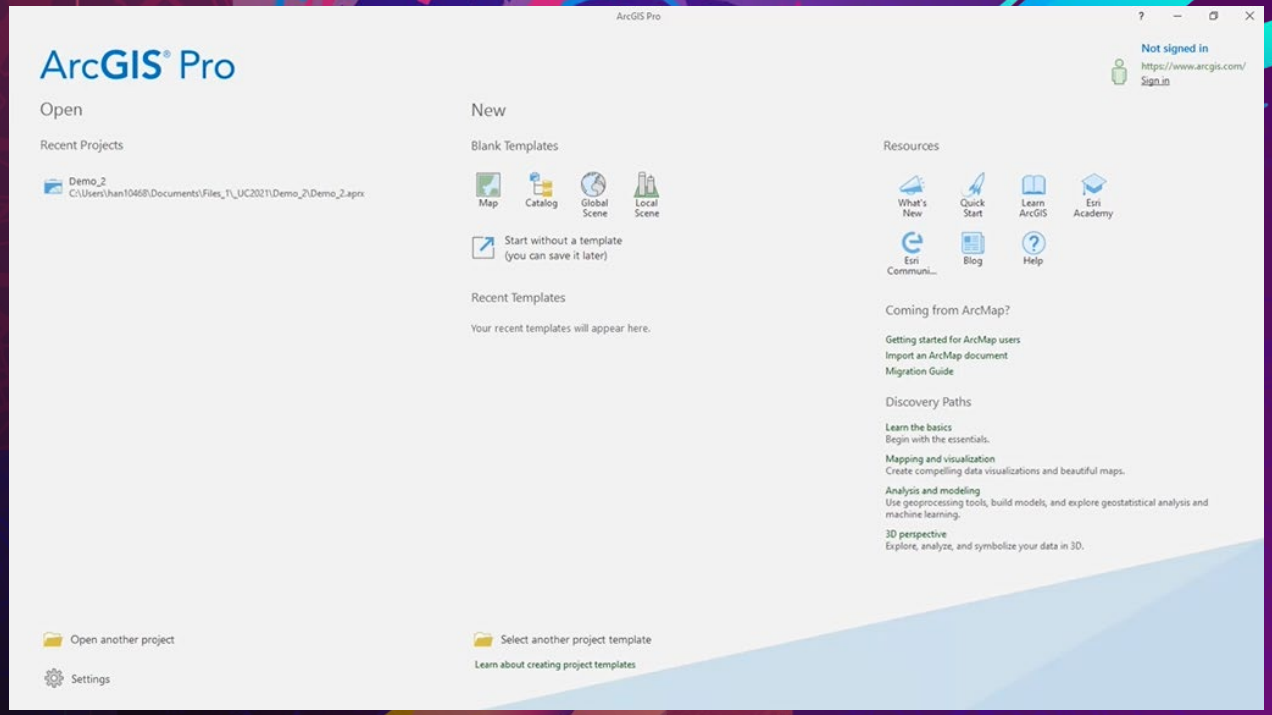
- build rich familiar dialog
  - interactive dialog
  - messaging
  - progress bar
- deep integration with ArcGIS Pro/ArcMap app
  - active map's layers
  - add result to display
- see "ArcPy: Building Geoprocessing Tools " tech workshop



Adding python Packages

- **Conda Environments**
  - Default is read only
  - Clone to add a package
  - Access to 1600 + packages
  - Python.exe with everyone

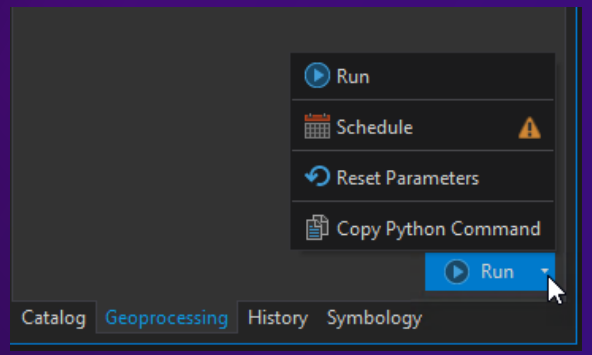


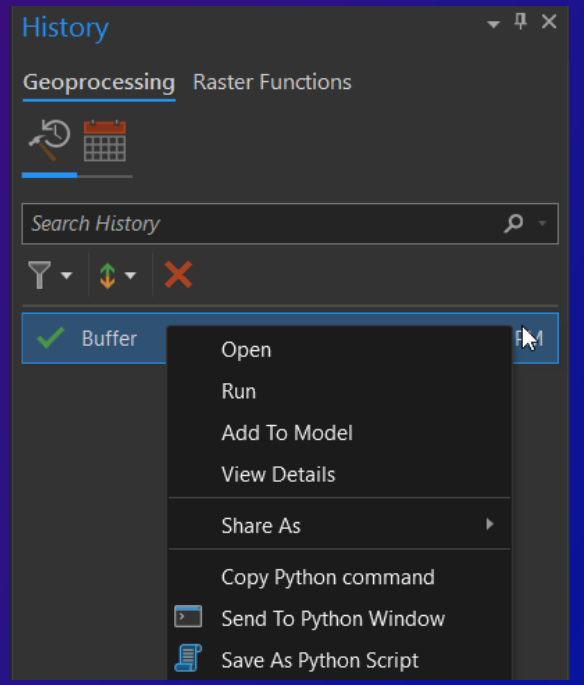


Demo : Adding and using Modules

Tips on getting started

- **Export Model Builder to Python**
- **Copy python command from tools**





Tips on getting started

- **Won't always work the first time**
  - **Write it, run it, break it**
  - **Get comfortable with tracebacks**

```
>>> x=2
>>> x+y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'y' is not defined
>>> y=3
>>> x+y
5
>>>
```

Other Python Technical Workshops at UC 2021

- ArcGIS API for Python: An Introduction
- ArcGIS: Python Overview
- Geospatial Deep Learning Using the ArcGIS API for Python

We want your feedback

Click on the Session Survey link
below this video window

Presentation Title

Presenter Names

2021 ESRI USER CONFERENCE

Section Header

Section Subhead

Demo Title

Presenter(s)