Using Python with ArcGIS

Jason Pardy (jpardy@esri.com)

Javier Abadia (javier.abadia@esri.es)

Agenda – A whirlwind tour

- Jason:
  - Python Essentials
  - Using Python in ArcGIS
  - Python Tools
  - Accessing Data
  - Map Automation
- Javier:
  - ArcGIS Server & Python
  - ArcGIS Online & Python
  - Other uses of Python

Python Essentials

Jason Pardy

What is Python?

- *“Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.” - python.org*

- “The” language for ArcGIS to perform data analysis, data management, data conversion, & map automation

Why use Python and ArcGIS?

- Automate repetitive tasks
- Build workflows that leverage hundreds of tools and functions
  - Geoprocessing services
- Extend ArcGIS
  - New tools
  - Desktop add-ins
- Administer services

Essentials – Python 101

- Logic for testing conditions
  - if, else statement
  - Operators: <, >, ==, not, in
- Iteration / looping control
  - For and while statements
- Key Python data structures
  - list, tuple, dictionary, set
- Building blocks
  - functions, modules, packages
- Python Standard Library / Built-ins
  - os, sys, math, datetime, collections, and many, many more!

Essentials - Extending Python

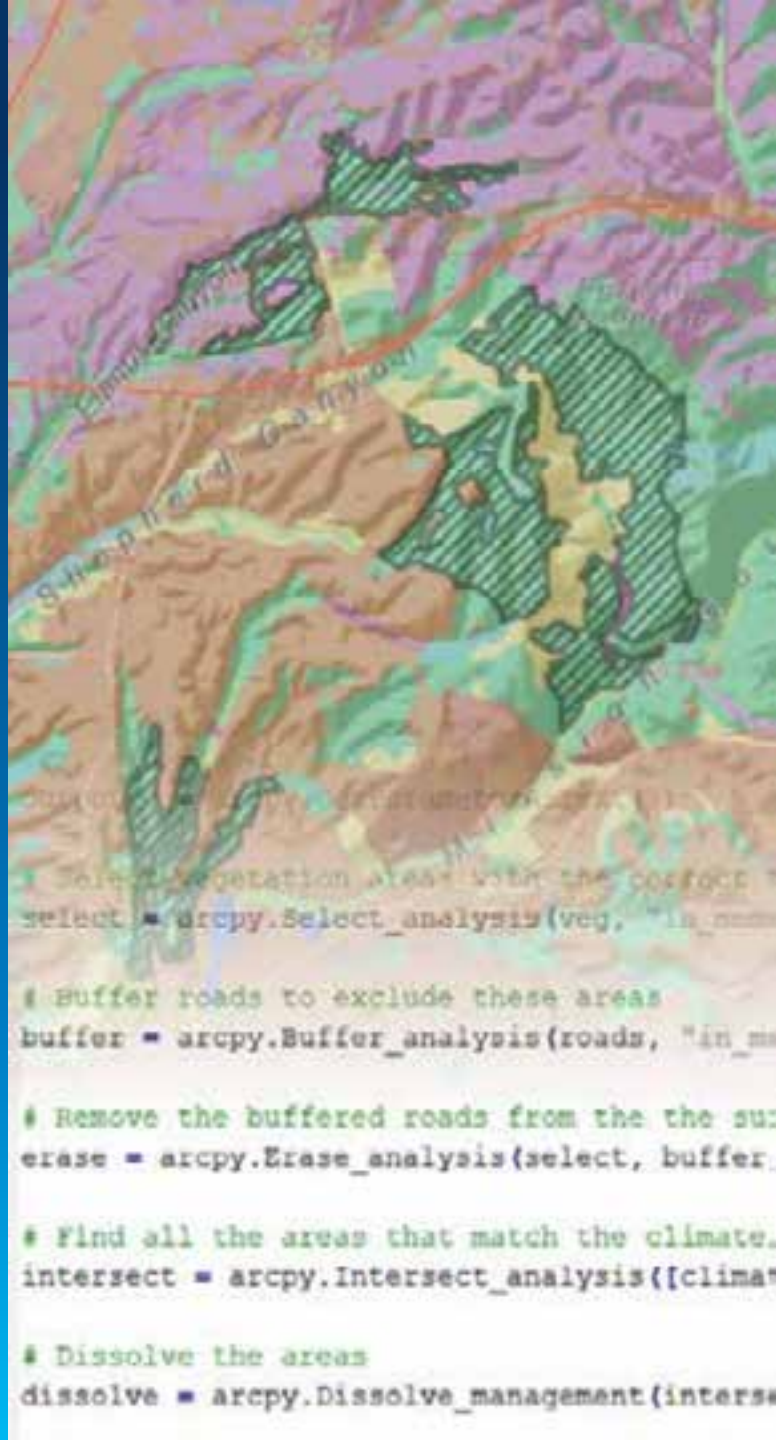
- Lot's of rich & powerful 3rd party libraries/packages
- Python has great tools for easily installing 3rd party packages:
  - easy_install
  - pip

INSTALLING PIP ON WINDOWS

- Download the latest installer for windows that fits your installed Python version: (download the exe at the bottom of http://pypi.python.org/pypi/setuptools).
- Install it.
- Add c:\Python2x\Scripts to the Windows path (replace Python2x with the correct directory name)
- Open a new (!) DOS prompt. From there run easy_install pip
- For the 64bit OS, see this answer at: http://stackoverflow.com/a/9038397/362951

Using Python in ArcGIS

Jason Pardy



ArcPy

- Python site-package included with ArcGIS
  - Access point to 900+ geoprocessing tools
  - Includes modules covering all areas of ArcGIS:
    - Data Access
    - Mapping
    - Extensions (sa, na)
    - Time
- Includes classes and functions making it easier to create objects such as spatial references, geometries, etc.

Multiple ways to Run Python

- Python window
- Python IDE
  - Blog: Choosing the right Python IDE
- Command prompt
  - Blog: **Schedule a Python script to run**
- Geoprocessing Tool
  - Script tool
  - Python toolbox – new @ 10.1

Demo

Using Python in ArcGIS

Find deepest points in a raster

Script tools

Jason Pardy

Why we create tools

- Easy to share
  - Generic
    - Can be used with different data and varied scenarios
- Communicates with the application
  - Layers from the map
  - Messages
  - Geoprocessing settings/environments
- Becomes part of the geoprocessing framework
  - Run from a tool dialog, ModelBuilder, Python
  - Can be shared as GP service or package

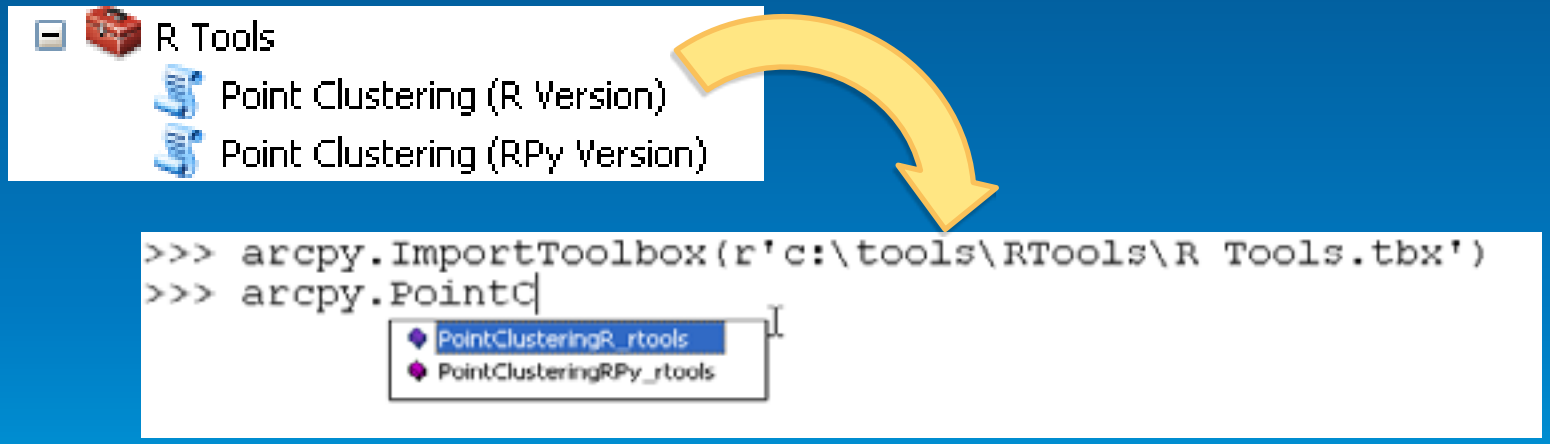
Demo

Creating Tools

Script tool & Python toolbox

Geoprocessing Tools

- Any tool, once created, can be called in Python by using the arcpy.ImportToolbox function
  - Creates tool wrappers for your toolbox



Tool Messages

- Executing a tool will produce 3 types of messages.
  - Informative messages (severity = 0)
  - Warning messages (severity = 1)
  - Error messages (severity = 2)

```
# start try block
try:
    arcpy.analysis.Buffer("c:/ws/roads.shp", "c:/outws/roads10.shp", 100)

# If an error occurs when running a tool, print the tool messages
except arcpy.ExecuteError:
    print arcpy.GetMessages(2)

# Any other error
except Exception as e:
    print e
```

Environments

- Script writers set the environment and tools use them
  - General settings
    - Current Workspace, Output Coordinate System, Extent
  - Raster analysis settings
    - Cell Size, Mask
  - Many more

arcpy.env.workspace

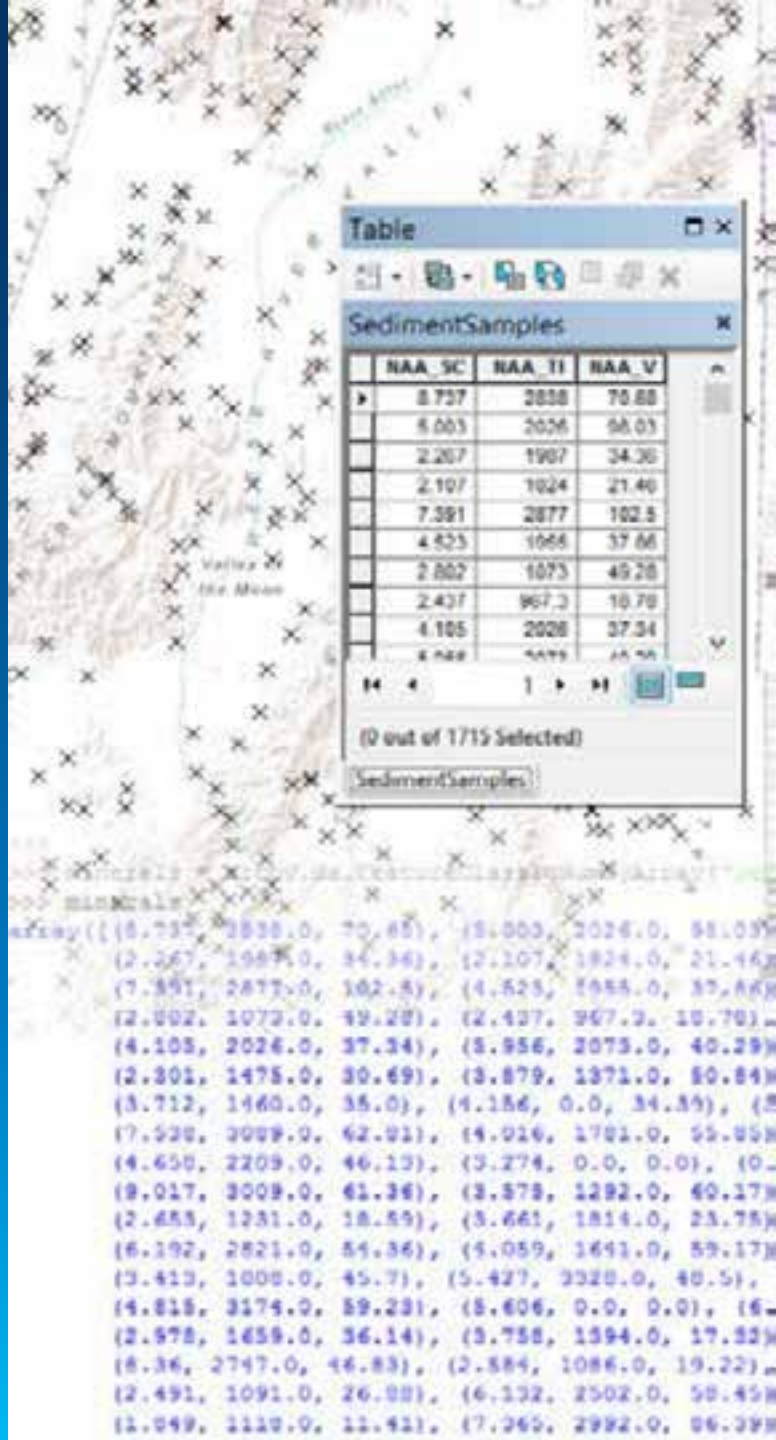
arcpy.env.outputCoordinateSystem

arcpy.env.extent

arcpy.env.cellSize

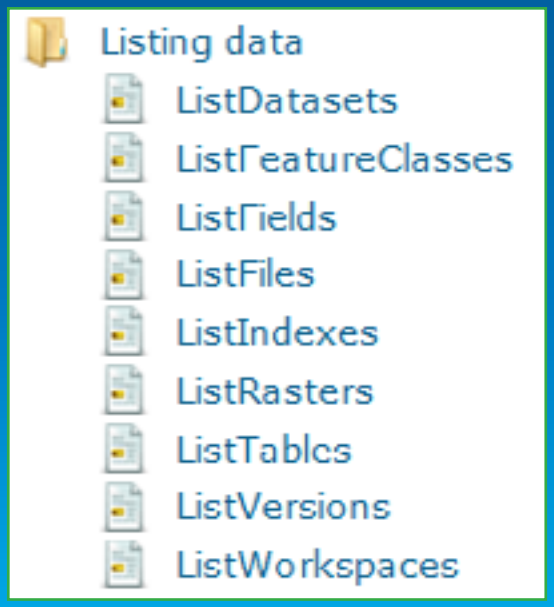
Accessing Data

Jason Pardy



Listing Data

- List functions exist to support these types of tasks:
  - Converting from one format to another (CAD to GDB)
  - Clipping a set of feature classes with a study area
  - Spill Modeling/Land use studies, etc.



Describe Function

- Allows script to determine properties of data
  - Data type (shapefile, coverage, network dataset, etc)
  - Shape type (point, polygon, line, etc)
  - Spatial reference
  - Extent of features
  - List of fields
- Returns an object with dynamic properties
- Logic can be added to a script to branch based on data properties

Data Access module (new @ 10.1)

- Improved cursor support (faster performance)
- Control of the edit session, edit operation
- Functions for converting tables/feature classes to and from NumPy arrays
- Support for versioning, replicas, domains, and subtypes
- Walk function (similar to os.walk but for ArcGIS data types)

Data Access cursors

- Much faster
- Supports *with* statements (no *del* needed)
- No need to access the full geometry

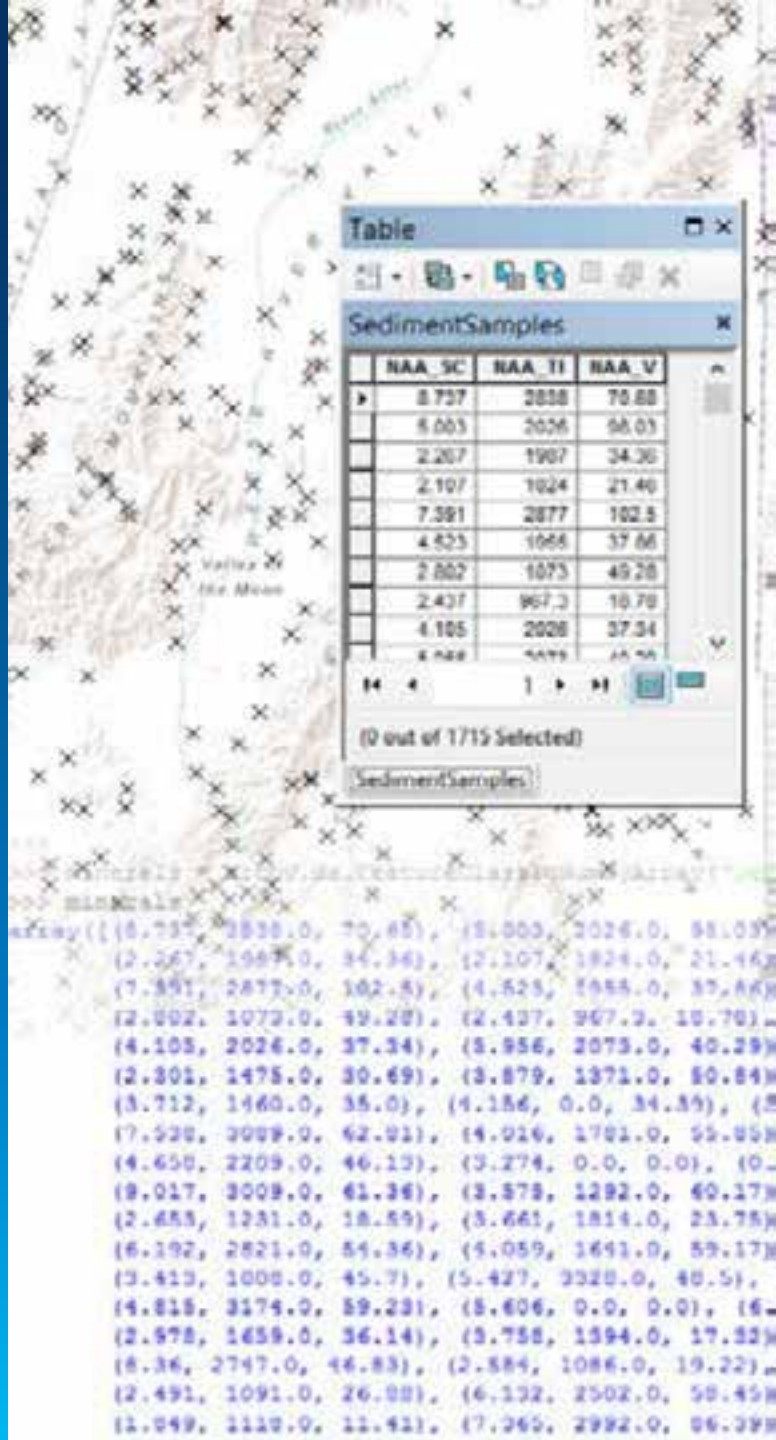
At 10.1

```
import arcpy

# Print the WELL_ID, WELL_TYPE, and the feature's X,Y.
fields = ["WELL_ID", "WELL_TYPE", "SHAPE@XY"]
with arcpy.da.SearchCursor("c:/data/base.gdb/well" , fields) as cursor:
    for row in cursor:
        print("{0}, {1}, {2}".format(row[0], row[1], row[2]))
```

Demo

Cursors

Jason Pardy



Geometry and cursors

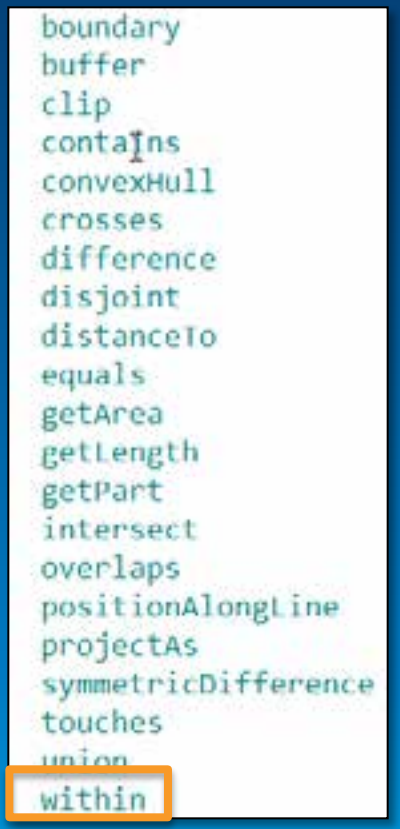
- Can create geometry in different ways
  - Geometry objects
  - List of coordinates
  - Using other formats
    - JSON, WKT, WKB

```
line = arcpy.Polyline(
    arcpy.Array([arcpy.Point(2,3),
                 arcpy.Point(3,5)]))

line = [(2,3), (3,5)]

line = {
    "type": "LineString",
    "coordinates": [[2, 3], [3,5]]}

cursor.insertRow([line])
```

Working with geometry

- Relational:
  - Is a point within a polygon?

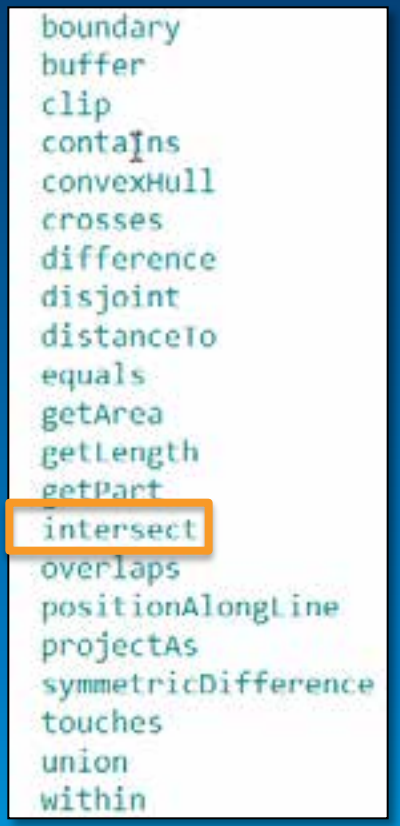
```
point.within(polygon)
```



Working with geometry

- Topological
  - What is the intersection of two geometries?

```
poly1.intersect(poly2)
```



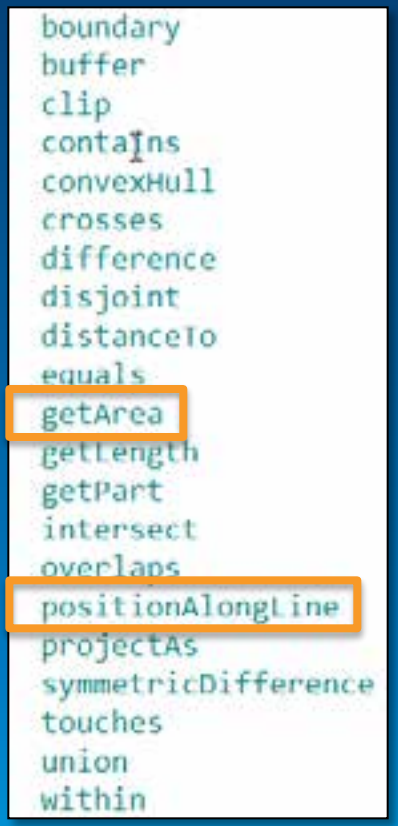
Working with geometry

- Others
  - What is the halfway point of a line?

```
line.positionAlongLine(0.5, True)
```

  - What is the geodesic area of a polygon?

```
poly.getArea('GEODESIC')
```



Demo

Topological Operation

Splitting Polygons

Map Automation & Map Production

Jason Pardy

Mapping module (arcpy.mapping)

- Contains functions and classes used to automate mapping tasks
  - Manage map documents and layers
  - Fix broken data sources
  - Update layer symbology across many maps
  - Export and print map documents
  - Automate map production / map series (create map books)

- Download Sample ArcPy Mapping Tools

Demo

arcpy.mapping

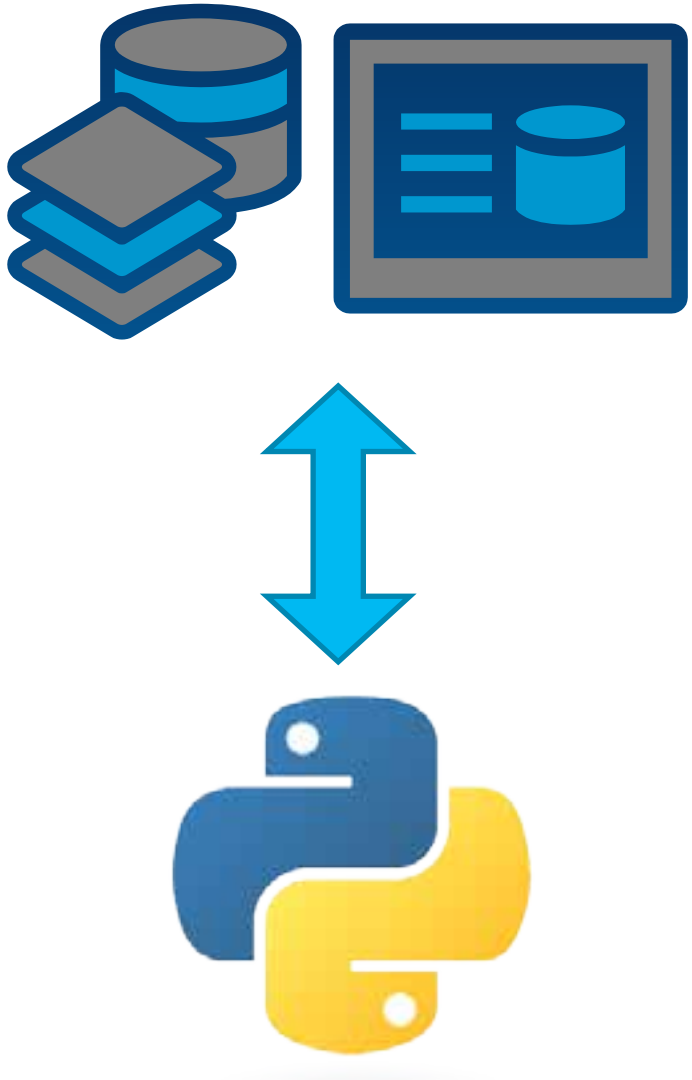
Update data sources, sql expressions, and label expressions

Recommended videos:

http://video.esri.com/watch/2935/python-map-automation-_dash_-introduction-to-arcpymapping
http://video.esri.com/watch/2934/python-map-automation-_dash_-beyond-the-basics-of-arcpymapping

ArcGIS Server & Python

Javier Abadía



ArcGIS for Server – REST API

- Services REST API

```
http://[server]/arcgis/rest/services
```

- Admin REST API

```
http://[server]/arcgis/admin
```

What can you do?

Anything

- clusters, machines
- datastores
- folders, services
- logs
- users, roles, permissions
- configuration backup
- …

http://resources.arcgis.com/en/help/main/10.1/index.html#//0154000005r1000000

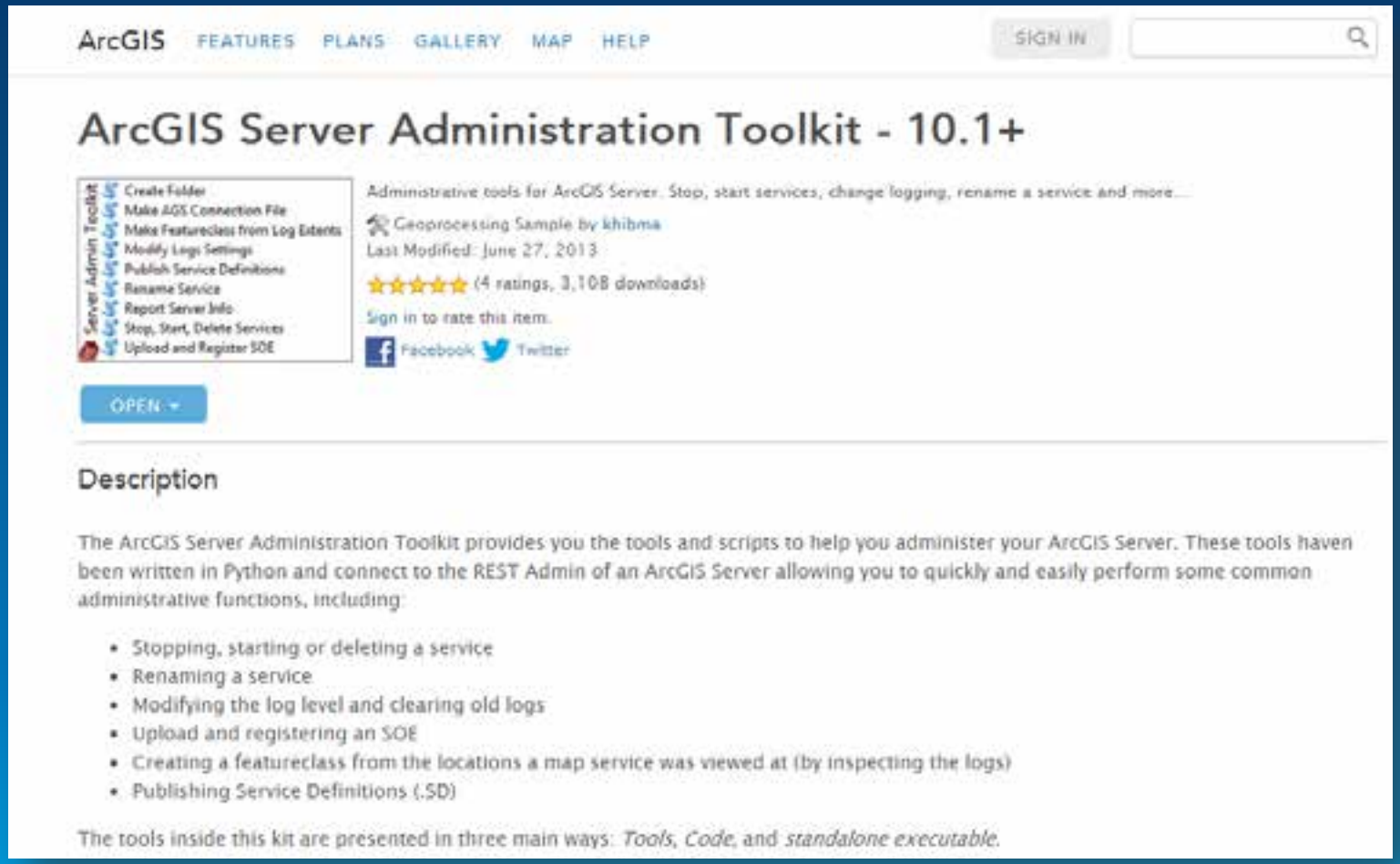
Steps

1. Generate a token
   - and keep it for subsequent requests
2. Issue REST requests
   - include the f=pjson parameter
   - get responses in JSON
   - use the correct HTTP method (GET, POST)
   - stateless

Python talks REST

- as any other language
- urllib/urllib2
  - send/receive http messages
- requests
  - better lib for sending/receiving http
  - pip install requests
  - http://www.python-requests.org

ArcGIS Server Administration Toolkit



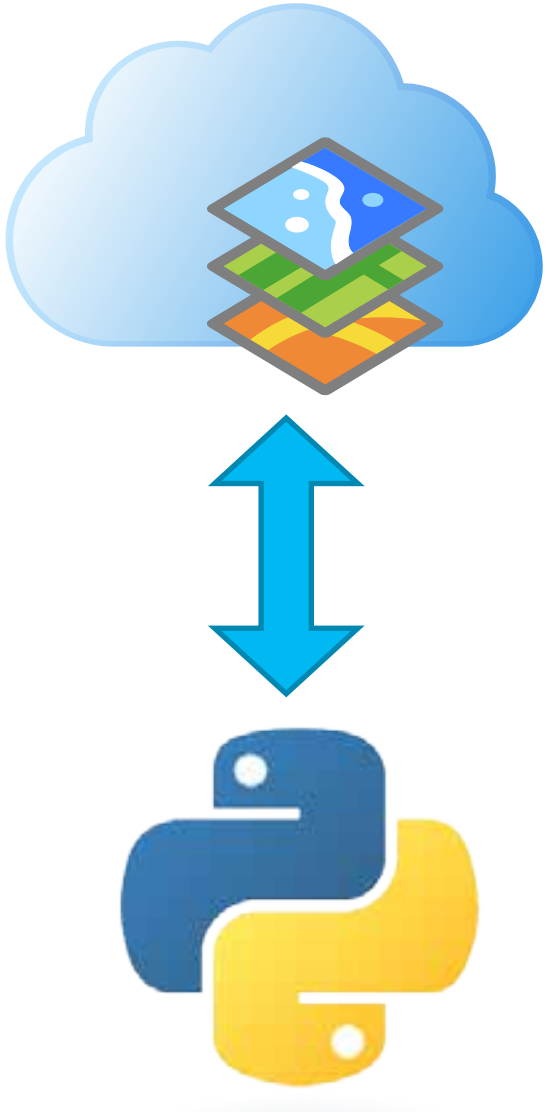
http://www.arcgis.com/home/item.html?id=12dde73e0e784e47818162b4d41ee340

Demo

ArcGIS for Server Admin Toolkit

ArcGIS Online & Python

Javier Abadía



ArcGIS Online

Can you guess? There's a REST API too!

- Map Services (ESRI provided)
- Task Services (ESRI provided)
- ArcGIS Online hosted services

- Portal Services
  - accessing/modifying Users, Groups, and Items

http://resources.arcgis.com/en/help/arcgisonline/index.html#/About_developing_with_portal_services/010q000000mv000000/

What can you do?

- Manage the Portal/ArcGIS Online organization
  - on behalf of authenticated user
- Search
- Groups
  - create, modify, delete, manage invitations, etc
- Users
  - modify, delete, manage group membership
- Items
  - upload, delete, reasign, update, group assignment, comments & ratings…

http://resources.arcgis.com/en/help/arcgis-rest-api/index.html#//02r3000000mt000000

Use cases

- Customized search
  - multi-portal
- Content Reports
  - list of users
  - list of all services used in webmaps
  - ensure quality of metadata
- Bulk operations
  - update service urls
  - group emailer
  - transfering ownership
  - register ArcGIS Server services into ArcGIS Online
- Backup

https://github.com/Esri/ago-admin-wiki

Other uses of Python

Javier Abadía

Python is amazing

ArcGIS is powerful

- many, many 3rd party libraries to do anything
  - read/write Excel files
    - (real .xls and .xlsx files not .csv)
  - scrap content from web pages
    - parsing the DOM
  - access databases, connect to Nintendo Wii remotes…

- you can use them together with ArcPy and REST API calls

amazing + powerful = ***awesome***

Reading/Writing excel workbooks

xlrd, xlwt

```
from xlrd import open_workbook

wb = open_workbook('simple.xls')

for s in wb.sheets():
     print 'Sheet:',s.name
     for row in range(s.nrows):
         values = []
         for col in range(s.ncols):
             values.append(s.cell(row,col).value)
         print ','.join(values)
     print
```

http://www.python-excel.org/

Scraping web content

BeautifulSoup

```
response = requests.get(url,params={'Id_Complejo': id_complejo})
response.encoding = 'utf-8‘
soup = bs(response.text)

…

root = soup.find('li',class_='resumen')
if root:
    cod_centro = int(root.find('div',class_='right').string)
    latitude = float(root.find('div',id='ctl00_ContentPlaceHolder1_...
    longitude = float(root.find('div',id='ctl00_ContentPlaceHolder1_...
    …
    print “%d,%.5f,%.5f,%s,%s,%s,%s" % (cod_centro,latitud,longitud,…
```

http://www.crummy.com/software/BeautifulSoup/bs4/doc/

Resources

- Python resource center
  - pro.arcgis.com/analysis/python/
- ArcPy Café
  - arcpy.wordpress.com
- Follow us on Twitter
  - @arcpy