Python – Beyond the Basics
David Wynne, Ken Smith
Agenda

1. Data structures & functions
2. Classes
3. Script tools
4. Cursors
5. Geometry
Python 201:
Data structures & functions
Key Python data structures

• Lists
  - Flexible
  - Ordered

• Tuples
  - Immutable
  - Ordered

• Dictionary
  - Key/value pairs
List comprehension

• Compact way of mapping a list into another
Defining Functions

• Organize and re-use functionality

```python
import arcpy

def increase_extent(extent, factor):
    """Increases the extent by the given factor""
    XMin = extent.XMin - (factor * extent.XMin)
    YMin = extent.YMin - (factor * extent.YMin)
    XMax = extent.XMax + (factor * extent.XMax)
    YMax = extent.YMax + (factor * extent.YMax)
    return arcpy.Extent(XMin, YMin, XMax, YMax)

oldExtent = arcpy.Describe("boundary").extent
newExtent = increase_extent(oldExtent, .1)
```
Data structures
Script tools
Quadrat Analysis tool
Geoprocessing Framework and Python

• Tools can be called from Python
• Python code can be turned into a tool

• A script tool…
  - Looks and behaves like any system tool
  - Provides default validation
  - Has no UI programming
Script tools

• Tools have parameters
  - Use GetParameterAsText (or GetParameter) to get inputs

• Script tools support messages
  - arcpy.AddMessage
  - arcpy.AddWarning
  - arcpy.AddError
### Script Template

```python
# Name:  template.py
# Purpose: 
# Author: 
# Created:  12/07/2012
# Copyright:  (c) company name
# ArcGIS Version:  10.1
# Python Version:  2.7
#
import os
import sys
import arcpy

def main(*argv):
    """TODO: Add documentation about this function here""
    try:
        #TODO: Add analysis here
        pass
    except arcpy.ExecuteError:
        print arcpy.GetMessages(2)
    except Exception as ee:
        print ee.args[0]
    # End main function

# Ensures this file is usable as a script, script tool, or an importable module
if __name__ == '__main__':
    argv = tuple(arcpy.GetParameterAsText(i) for i in range(arcpy.GetArgumentCount()))
    main(*argv)
```

[http://esriurl.com/4558](http://esriurl.com/4558)
Quadrat Analysis tool

Part B:
Jump start your analysis with a template
arcpy classes
arcpy classes

- Classes are used to create objects
- Frequently used for...
  - Tool parameters
  - Working with geometry
arcpy Classes

• Most tool parameters are easily defined with a string or number

• Some are not:
  - Spatial reference
  - Field map

• Classes can be used to define these parameters
Using classes for parameters

- **Extent**

```python
extent = arcpy.extent(-117.1, 14.5, -86.7, 32.7)
arcpy.CreateRandomPoints_management(
    arcpy.env.workspace,
    'samplepoints',
    constraining_extent=extent
)
```

- **SpatialReference**

```python
arcpy.CreateFeatureclass_management(
    arcpy.env.workspace, 'hydrology', 'POLYGON',
    spatial_reference=arcpy.SpatialReference(32145)
)
```
Quadrat Analysis tool

Part C:
- Extent class
- Cursors
- Chi-square function
Recap

1. Fast development
   I. A script template gets you started
   II. Easy access to tools and functions through arcpy
   III. No UI development required
   IV. Script tools provide default validation
Recap…

2. Used a function to define / group useful functionality
3. Used open source libraries (numpy)
4. Easy to share and deploy
   - No ArcObjects or dlls to register
Cursors
Cursors

• Use cursors to access records and features

<table>
<thead>
<tr>
<th>Cursor Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchCursor</td>
<td>Read-only</td>
</tr>
<tr>
<td>UpdateCursor</td>
<td>Update or delete rows</td>
</tr>
<tr>
<td>InsertCursor</td>
<td>Insert rows</td>
</tr>
</tbody>
</table>

• Two varieties
  - ‘Classic’ cursors
  - ‘Data access’ cursors (new at 10.1)
Cursor mechanics

• Data access cursors use lists and tuples
  - Values are accessed by index

```python
• cursor = arcpy.da.InsertCursor(table, ["field1", "field2"])
• cursor.insertRow([1, 10])
```

• Classic cursors use row objects
  - Values are accessed by setValue/getValue

```python
• cursor = arcpy.InsertCursor(table)
• row = cursor.newRow()
• row.setValue("field1", 1)
• row.setValue("field2", 10)
• cursor.insertRow(row)
```
With statements

- arcpy.da Cursors support with statements

```python
with arcpy.da.SearchCursor(table, field) as cursor:
    for row in cursor:
        print row[0]
```
Cursor performance

- Use only those fields you need
- Use tokens
  - Get only what you need
  - *Full geometry is expensive*
arcpy.da cursors
Geometry
Geometry and cursors

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

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

• Relational:
  - Is a point within a polygon?

```python
point.within(polygon)
```
Working with geometry

- Topological
  - What is the intersection of two geometries?

```python
poly1.intersect(poly2)
```
Working with geometry

• Others
  - What is the halfway point of a line?
    ```python
    line.positionAlongLine(0.5, True)
    ```
  - What is the geodesic area of a polygon?
    ```python
    poly.getArea('GEODESIC')
    ```
Resources

• Python resource center
  - pro.arcgis.com/analysis/python/

• ArcPy Café
  - arcpy.wordpress.com

• Python at UC 2013
  - esriurl.com/uc13python
Thank you…

Please fill out the session evaluation

First Offering ID (Tues.): 1210
Second Offering ID (Wed.): 1311

Online – www.esri.com/ucsessionsurveys
Paper – pick up and put in drop box