ArcPy Tips & Tricks

Clinton Dow – Geoprocessing Product Engineer @ Esri

Tip #1 – ArcPy in an IDE

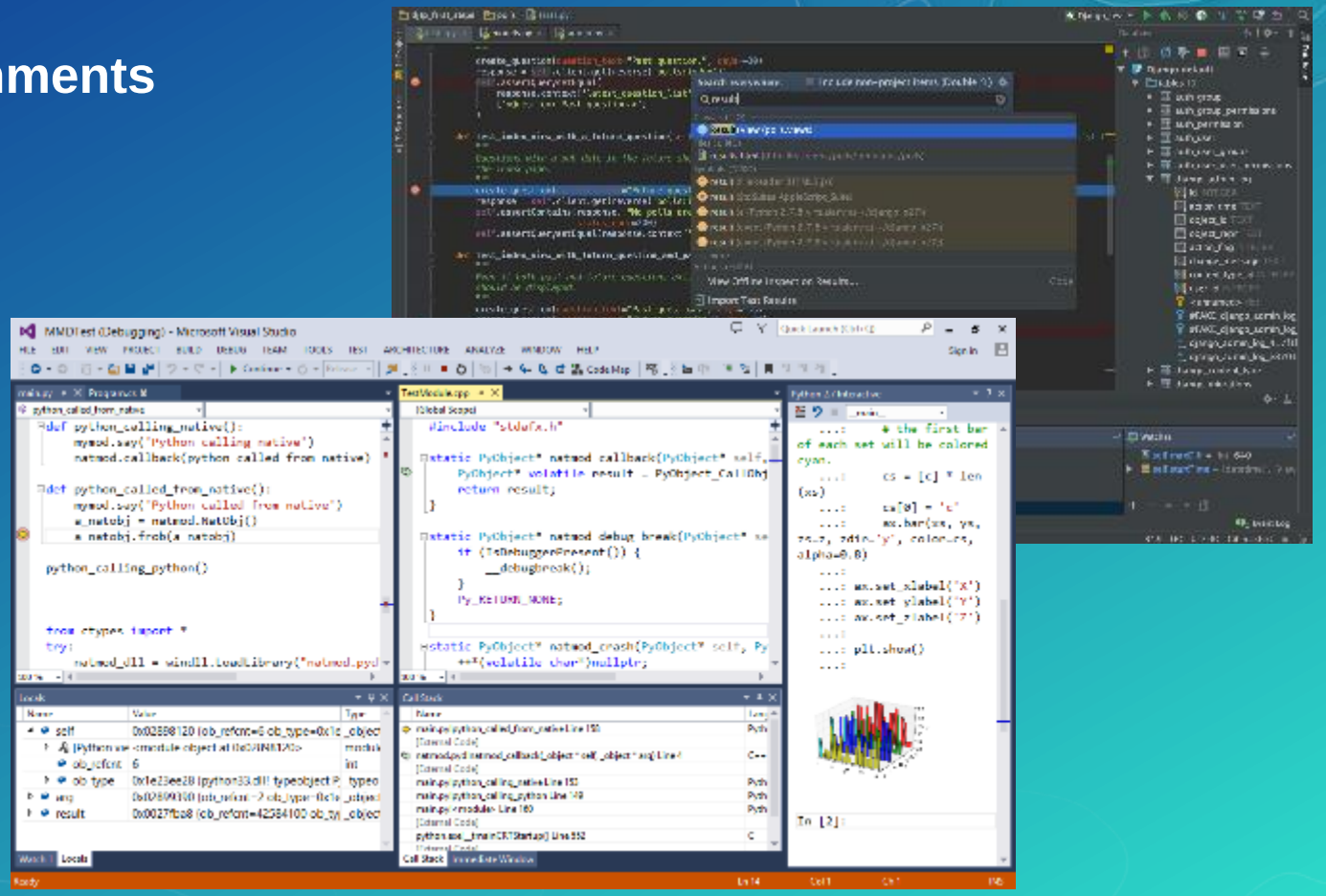
GIS from the comfort of your development environment

- **Recommended IDEs**
  - PyCharm
  - Python Tools for Visual Studio
  - Spyder
  - Eclipse with PyDev
  - Wingware

Tip #1 – ArcPy in an IDE

GIS from the comfort of your development environment

- **Integrated Development Environments**
  - **Debuggers**
  - **Test Frameworks**
  - **Coverage**
  - **PyLint/pycodestyle**
  - **Version Control Integration**
  - **Refactoring Tools**
  - **Virtual Environment Support**
  - **AutoComplete**



Tip # 2 – Decouple Your Tool Logic

Toolboxes, Python Toolboxes, Interpreters, Command Line, Services…

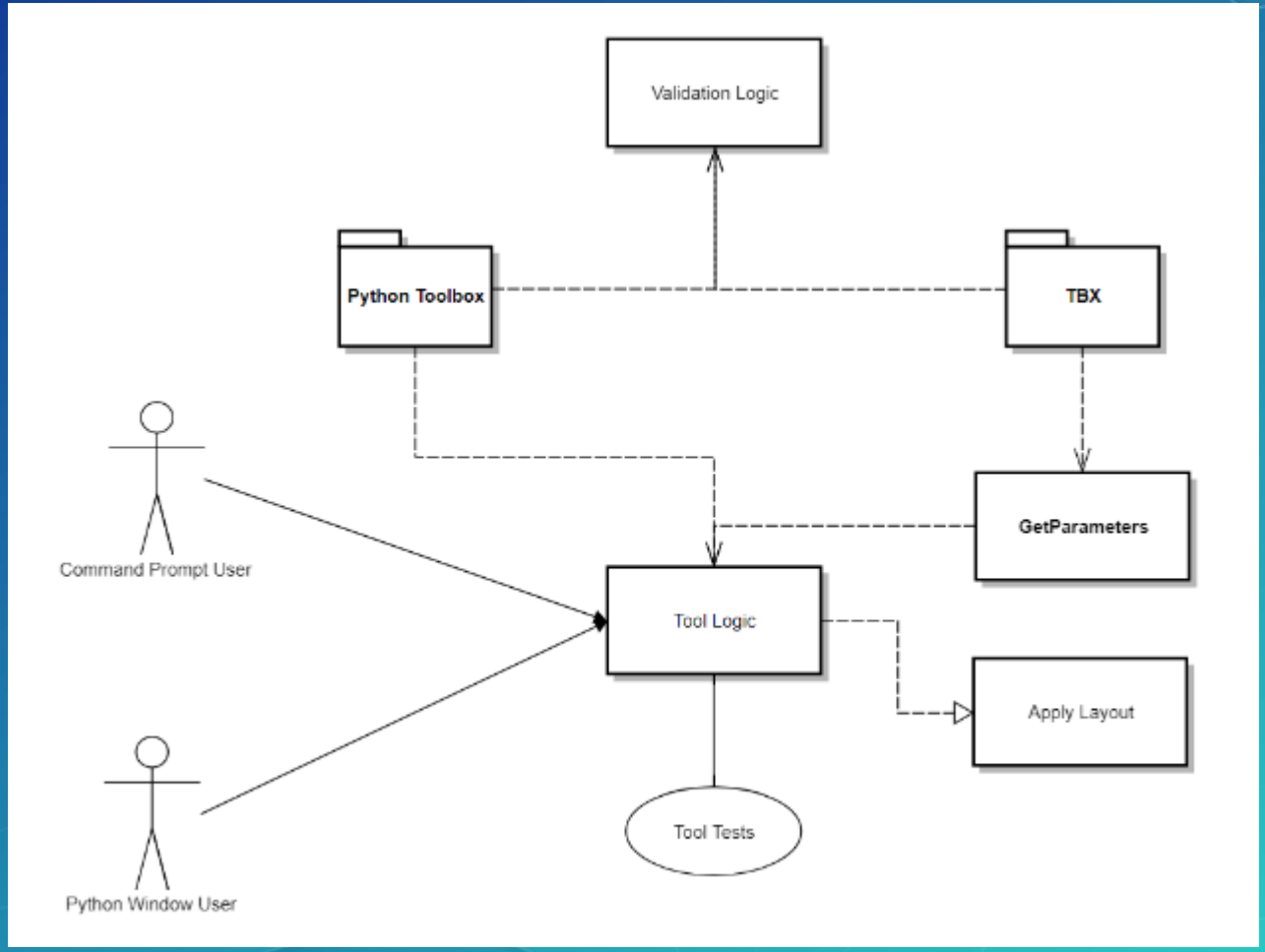
- **Multiple ways to create tools**
  - ArcGIS Toolbox (.tbx)
  - Python Toolbox (.pyt)
  - Stand-alone Scripts (.py)
  - Interactive Python Window
- **Test Driven Development**
- **Distutils to create packaged code**

  Sharing

  Install into tools as needed

Decoupled ArcPy Tool Logic Diagram

Modularized Code



Tip #3 – There Must Be A Better Way!

Never Underestimate The Python Standard Library!

- **Raymond Hettinger**
  - **Transforming Code Into Beautiful, Idiomatic Python**
    - https://www.youtube.com/watch?v=OSGv2VnC0go
  - **Beyond PEP 8 – Best Practices for Beautiful, Intelligible Code**
    - https://www.youtube.com/watch?v=wf-BqAjZb8M

Tip #4 – Comprehensions with Cursors

Using Pure Python Data Types to Work with Feature Classes

```
import arcpy

data_path = r"C:\Path\to\feature\class"


def cursor_list_comprehension(path):
    l = [row for row in arcpy.da.SearchCursor(path, ["SHAPE@", "Language"])]
    return l


def cursor_dictionary_comprehension(path):
    d = {row[0]: row[1:] for row in arcpy.da.SearchCursor(path, ["OID@", "MsgText", "Language"])}
    return d


def cursor_generator_comprehension(path):
    g = (row for row in arcpy.da.SearchCursor(path, ["SHAPE@", "Language"]))
    return g


def cursor_generator_statement(path):
    with arcpy.da.SearchCursor(path, ["OID@", "MsgText"]) as cursor:
        for row in cursor:
            yield str(row[0])
```

Tip #5 – Filter & Map with Comprehensions

Using Pure Python Data Types to Work with Feature Classes

```
import arcpy

data_path = r"C:\Path\to\feature\class"


def cursor_comprehension_map(path):
    # Map function
    # m = map(lambda x: x[0] + 1.0, [row for row in arcpy.da.SearchCursor(path, ['OID@'])])

    m = [row[0] + 1 for row in arcpy.da.SearchCursor(path, ['OID@'])]
    return m


def cursor_comprehension_filter(path):
    # Filter function
    # f = filter(lambda x: x ==1, [row for row in arcpy.da.SearchCursor(path, ['OID@'])])

    f = [row for row in arcpy.da.SearchCursor(path, ['OID@']) if row[0] == 1]
    return f
```

Tip #6 – Collections Example

Using Pure Python Data Types to Work with Feature Classes

```
from collections import Counter
import arcpy

data_path = r"C:\Users\clin8331\Desktop\Temp\tw.gdb\tw_lang_europe"


def count_field_occurances(data_path):
    c = Counter()
    with arcpy.da.SearchCursor(data_path, ["OID@", "Language"]) as cursor:
        for row in cursor:
            c[row[1]] += 1
    return c
```

Tip #7 – Use Third Party Packages

- **Anaconda**
  - Curated collection of popular Python packages
  - Handles dependencies
  - Integrated into ArcGIS Pro
- **Pip**
  - Python Package Index (PyPI)
  - Over 100k Packages
  - Works with 10.x and Pro

Name	Version	Summary / License
affine	2.0.0	Matrices describing affine transformation of the plane. / BSD 3-clause
alabaster	0.7.10	Configurable, Python 2+3 compatible Sphinx theme / BSD
alpaca_static	1.5.17	Easy Forms for jQuery / Apache License 2.0
anaconda-clean	1.1.0	Delete Anaconda configuration files / BSD
anaconda-client	1.6.3	anaconda.org command line client library / BSD 3-clause
anaconda-navigator	1.6.2	Anaconda Navigator / proprietary - Continuum Analytics, Inc.
anaconda-project	0.6.0	reproducible, executable project directories / BSD 3-clause
anaconda-verify	1.3.7	tool for (passively) verifying conda recipes and conda packages / BSD
ansi2html	1.1.1	Convert text with ANSI color codes to HTML or to LaTeX. / General Public License v3.0
anyqt	0.0.8	PyQt4/PyQt5 compatibility layer. / GPL-3.0
appdirs	1.4.0	Python module for determining appropriate platform-specific dirs / MIT
appnope [osx]	0.1.0	Disable App Nap on OS X 10.9 / BSD
appscript [osx]	1.0.1	Control AppleScriptable applications from Python / Public Domain
apptools	4.4.0	Includes packages useful for creating applications / BSD

Tip #8 – Pathlib makes paths easy!

- **Object-Oriented Paths**
  - Native Python in 3.4+
  - Install via Pip in 2.7.x
- **Folder hierarchy as list**
  - Access parent folder as index 0
  - Parent of parent is index 1, etc.
- **File Comparison**
  - Compare contents of files easily

```
>>> p = PureWindowsPath('c:/foo/bar/setup.py')
>>> p.parents[0]
PureWindowsPath('c:/foo/bar')
>>> p.parents[1]
PureWindowsPath('c:/foo')
>>> p.parents[2]
PureWindowsPath('c:/')
```

Tip #9 - Help us make ArcPy better!

- GeoNet
  - http://geonet.esri.com
  - https://geonet.esri.com/community/developers/gis-developers/python
- Conda Support Email
  - conda@esri.com
- ArcPy Café
  - https://arcpy.wordpress.com/

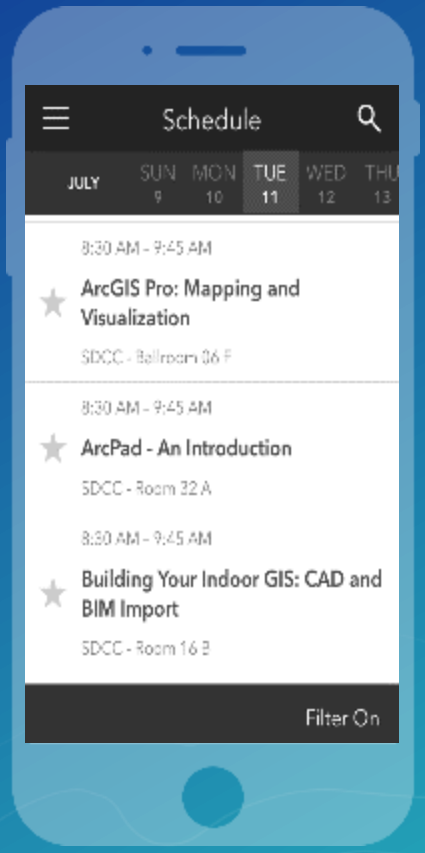
Conclusion

- IDEs are useful tools for aiding development
- The Python Standard Library is very powerful on its own
- Modularized code is your friend
  - Easier to follow
  - Reuse code
  - Debugging small units of functionality
- Interact with us!
  - cdow@esri.com

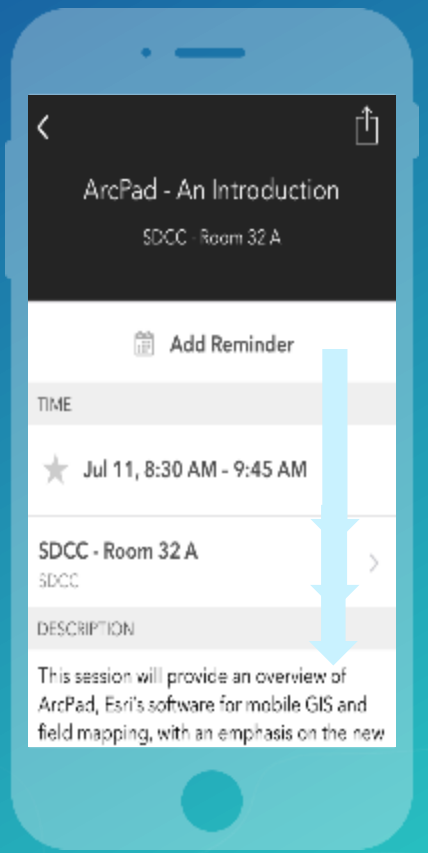
Please Take Our Survey on the **Esri Events App!**

Download the Esri Events app and find your event

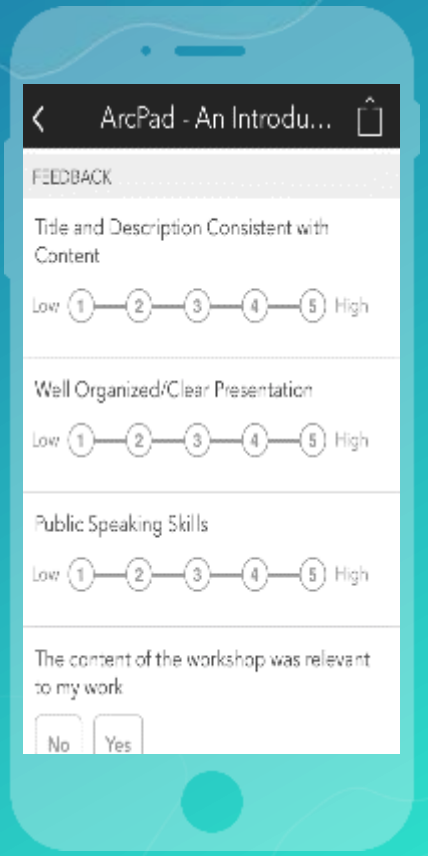
Select the session you attended



Scroll down to find the survey



Complete Answers and Select "Submit"



Clinton Dow – cdow@esri.com