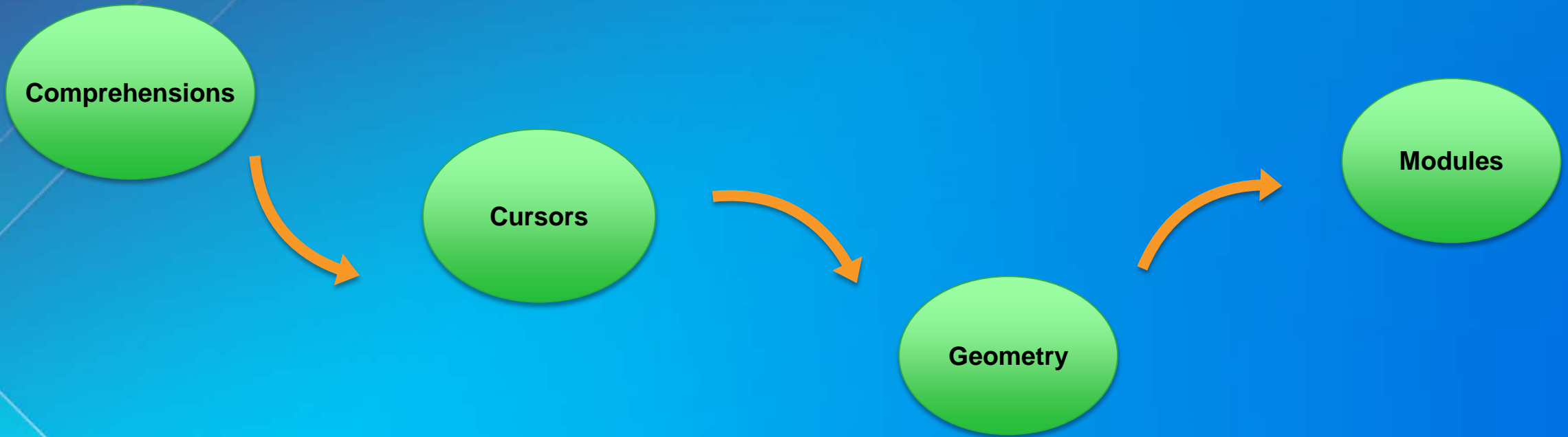




Python Beyond the Basics

Ghislain Prince

Jeff Bigos





Comprehensions

Comprehensions

- List Comprehensions
 - Loop and filter items from an existing list
 - List is returned

```
dist = range(5, 100, 5)  
analysisDist = [num for num in dist if num < 30]
```

Comprehensions

- List Comprehensions
 - Loop and filter items from an existing list
 - List is returned

```
dist = range(5, 100, 5)  
analysisDist = [num for num in dist if num < 30]
```

Comprehensions

- List Comprehensions
 - Loop and filter items from an existing list
 - List is returned

```
dist = range(5, 100, 5)  
analysisDist = [num for num in dist if num < 30]
```

Comprehensions

- List Comprehensions
 - Loop and filter items from an existing list
 - List is returned

```
dist = range(5, 100, 5)  
analysisDist = [num for num in dist if num < 30]
```

Comprehensions

- Dictionary Comprehensions
 - Loop and filter items from an existing Dictionary
 - Dictionary is returned

```
atbatDict = {kab: vab for kab, vab in battertargetABINFO[0].attrs.iteritems() if battertargetABINFO[0].attrs.has_key(kab)}
```


Comprehensions

- Dictionary Comprehensions
 - Loop and filter items from an existing Dictionary
 - Dictionary is returned

```
atbatDict = {kab: vab for kab, vab in battertargetABINFO[0].attrs.iteritems() if battertargetABINFO[0].attrs.has_key(kab)}
```

Comprehensions

- Dictionary Comprehensions
 - Loop and filter items from an existing Dictionary
 - Dictionary is returned

```
atbatDict = {kab: vab for kab, vab in battertargetABINFO[0].attrs.iteritems() if battertargetABINFO[0].attrs.has_key(kab)}
```

Cursors

Cursors

Type	Function
SearchCursor	Read-only
UpdateCursor	Update or delete rows
InsertCursor	Insert rows

- **Two varieties**
 - Arcpy.da cursors (10.1 onwards)
 - 'Classic' cursors

Understanding Data access Cursors

- Fields parameter
 - Index position in fields parameter defines value access
 - Provide needed fields

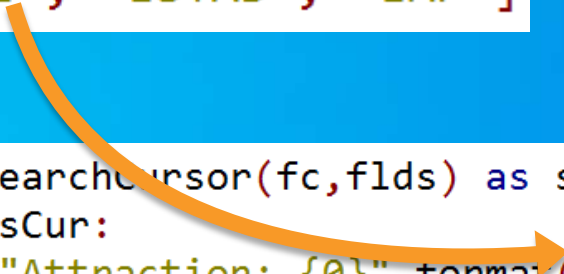
```
#           0           1           2  
flds = ["NAME", "ESTAB", "EMP"]
```

Understanding Data access Cursors

- Fields parameter
 - Index position in fields parameter defines value access
 - Provide needed fields

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

```
with arcpy.da.SearchCursor(fc, flds) as sCur:  
    for row in sCur:  
        print ("Attraction: {}".format(row[0]))
```



Understanding Data access Cursors

- Where Clause

- Filters the records returned

```
exp = """ ESTAB > 1950 """
```

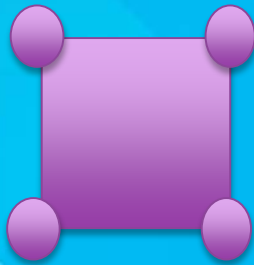
- Spatial Reference

- Project data on the fly

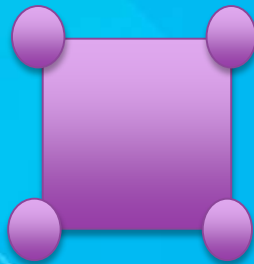
```
spatialRef = arcpy.SpatialReference(26711)  
spatialRef.create()
```



Geometry



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)  
]))
```

Creating Geometry

- List of coordinates
 - *coords

```
lineCoords = [[-22.570,63.968],[-22.573,63.969],[-22.575,63.968]]

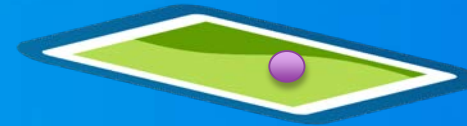
line = arcpy.Polyline(
    arcpy.Array([arcpy.Point(*coords) for coords in lineCoords])
)

print(line.length)
```

Geometry Operators

- **Relational:**
 - Is a point within a polygon
 - Return Boolean Values

```
pnt.within(Polygon)
```



Geometry Operators

- **Contains:**



- **Crosses:**



- **Touches:**



- **Additional Methods in the documentation**

The background features a vibrant blue gradient. On the left side, there are several overlapping geometric shapes: a large purple triangle pointing upwards, a yellow triangle pointing downwards, and a dark purple triangle pointing downwards. The yellow triangle contains a faint, light-colored map of a region, possibly a state or province. The word "Modules" is centered in the blue area.

Modules

Modules: Included with python

- **urllib: functions and classes for using internet resources**
 - **Urlencodings, POST, GET**
- **json: work with json data**
 - **Encoding, decoding**
- **zipfile: create and extract from zipfiles**
- **Resources:**
 - **Python help: Global Module Index**
 - **Python module of the week**

Modules: Included with ArcGIS

- **numpy: n- dimensional array**
 - Spatial Analyst, arcpy.da conversion functions
- **matplotlib: Graphing and Charting**
 - Spatial Statistics visual outputs
- **xlrd / xlwt: read and write excel files**
 - Used in the excel toolset
- **Resources:**
 - Python help: [Global Module Index](#)
 - Python module of the week



Understanding our world.