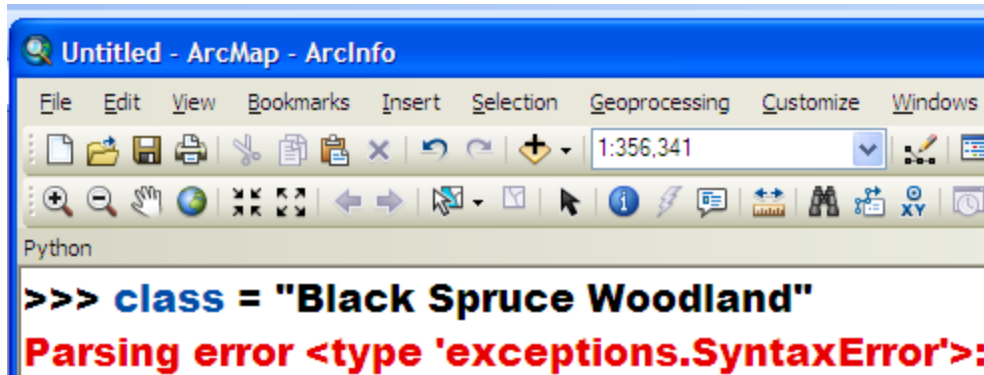
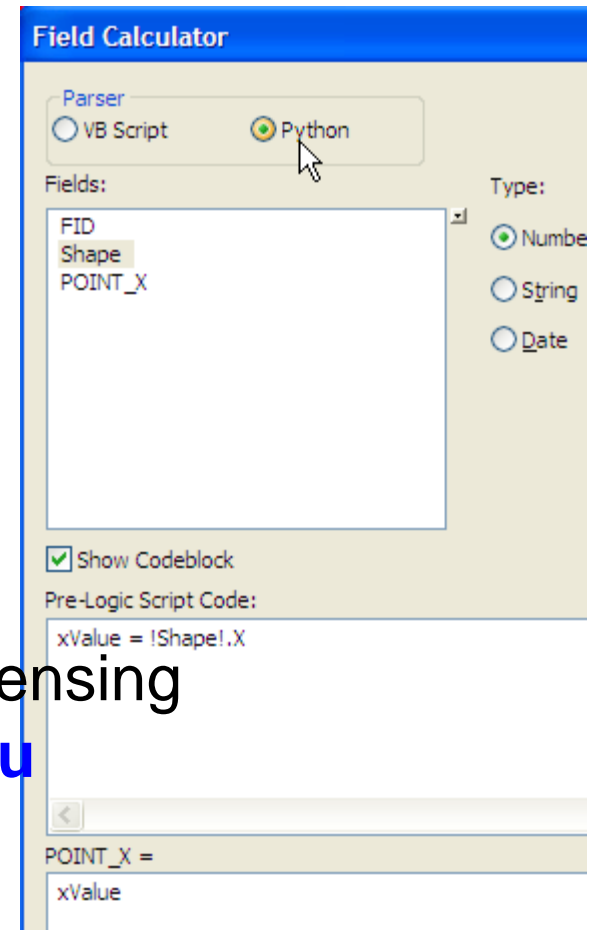


Python Scripting in ArcGIS10: 25 Potential Sources of Confusion



Dave Verbyla
Professor of GIS/Remote Sensing
dverbyla@alaska.edu















Why learn ArcGIS Python

- Python scripting language installs with ArcGIS
- Python window in ArcGIS
- Create your own script tools
- Arcpy.mapping (new with ArcGIS10)



Address  C:\AKtemperature















 ak.cru.temp.1.2000.txt	 ak.cru.temp.11.2010.txt
 ak.cru.temp.1.2001.txt	 ak.cru.temp.12.2000.txt
 ak.cru.temp.1.2002.txt	 ak.cru.temp.12.2001.txt

 ak.cru.temp.1
 ak.cru.temp.1
 ak.cru.temp.1
 ak.cru.temp.1

Address  C:\AKtemperature



Go

 Dec2005.txt	 Jan2010.txt
 Dec2006.txt	 Jul2000.txt
 Dec2007.txt	 Jul2001.txt
 Dec2008.txt	 Jul2002.txt
 Dec2009.txt	 Jul2003.txt
 Dec2010.txt	 Jul2004.txt
 Feb2000.txt	 Jul2005.txt

```

import os, glob
path = 'C:/AK_temperature'
os.chdir(path)
lstFiles = glob.glob('*.txt')
lstMonths = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Oct', 'Sep', 'Nov', 'Dec']
for File in lstFiles:
    if File[12:14] == '1.' :
        Month = lstMonths[0]
    elif File[12:14] == '2.' :
        Month = lstMonths[1]
    elif File[12:14] == '3.' :
        Month = lstMonths[2]
    elif File[12:14] == '4.' :
        Month = lstMonths[3]
    elif File[12:14] == '5.' :
        Month = lstMonths[4]
    elif File[12:14] == '6.' :
        Month = lstMonths[5]
    elif File[12:14] == '7.' :
        Month = lstMonths[6]
    elif File[12:14] == '8.' :
        Month = lstMonths[7]
    elif File[12:14] == '9.' :
        Month = lstMonths[8]
    elif File[12:14] == '10' :
        Month = lstMonths[9]
    elif File[12:14] == '11' :
        Month = lstMonths[10]
    else :
        Month = lstMonths[11]
    newName = Month + File[-8:]
    os.rename(File, newName)

```

```

#####
# Python script to rename #
# files from mmyyyy.txt to #
# MONyyyy.txt format #
#####
def strMonth(inString):
    lookup = {'1.': 'Jan', '2.': 'Feb',
              '3.': 'Mar', '4.': 'Apr', '5.': 'May',
              '6.': 'Jun', '7.': 'Jul', '8.': 'Aug',
              '9.': 'Sept', '10': 'Oct', '11': 'Nov',
              '12': 'Dec'}
    return lookup.get(inString, 'ERR')

import os, glob
try:
    os.chdir(r'C:\AKtemperature')
    txtFiles = glob.glob('*.txt')
    for oldName in txtFiles:
        strMon = strMonth(oldName[12:14])
        newName = strMon + oldName[-8:]
        os.rename(oldName, newName)
except:
    print "Script error..."

```



Source of Confusion

- Python interpreter is case sensitive

Interactive Window

```
>>> Current_directory = os.getcwd( )
```

```
>>> print Current_Directory
```

```
Traceback (most recent call last):
```

```
  File "<interactive input>", line 1, in <module>
```

```
NameError: name 'Current_Directory' is not defined
```

```
>>> Print Current_directory
```

```
Traceback ( File "<interactive input>", line 1
```

```
    Print Current_directory
```

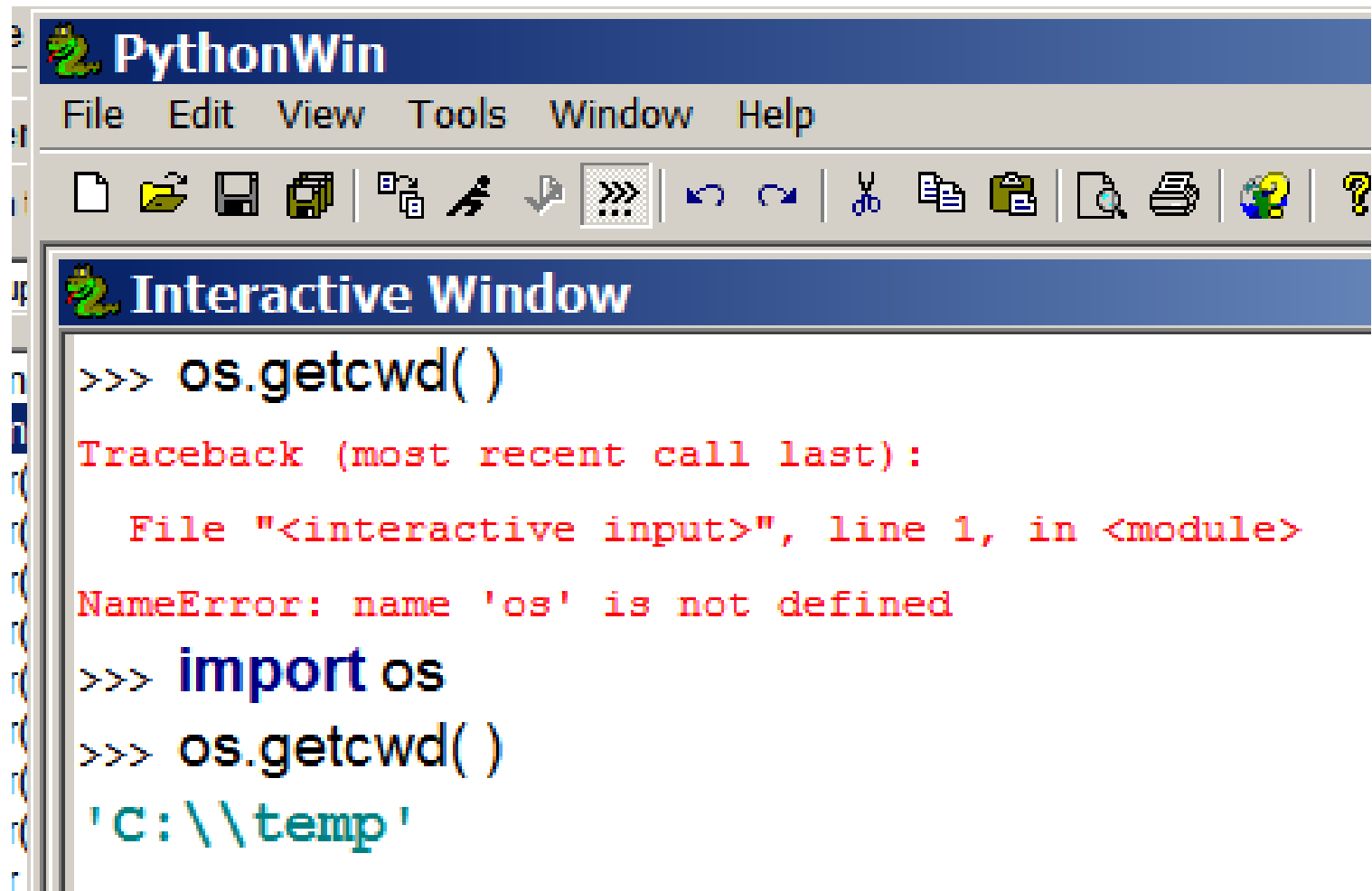
```
      ^
```

```
SyntaxError: invalid syntax
```

```
>>> print Current_directory
```

```
C:\temp
```

- Python command not understood without appropriate module loaded



The screenshot shows the PythonWin application window. The title bar reads "PythonWin". The menu bar includes "File", "Edit", "View", "Tools", "Window", and "Help". The toolbar contains various icons for file operations and execution. The main window is titled "Interactive Window" and displays the following text:

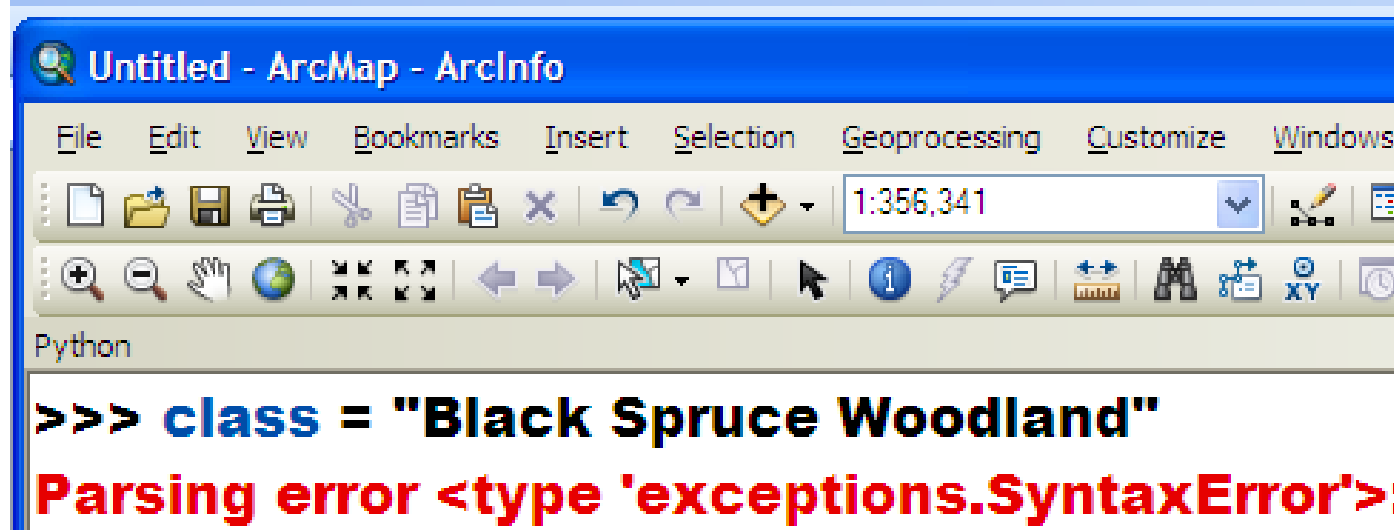
```
>>> os.getcwd( )  
Traceback (most recent call last):  
  File "<interactive input>", line 1, in <module>  
NameError: name 'os' is not defined  
  
>>> import os  
>>> os.getcwd( )  
'C:\\temp'
```



```
>>> import arcpy.mapping
>>> arcpy.mapping.MapDocument(r'c:\temp\map_book.mxd')
<MapDocument object at 0xf07670[0xf0c960]>
```

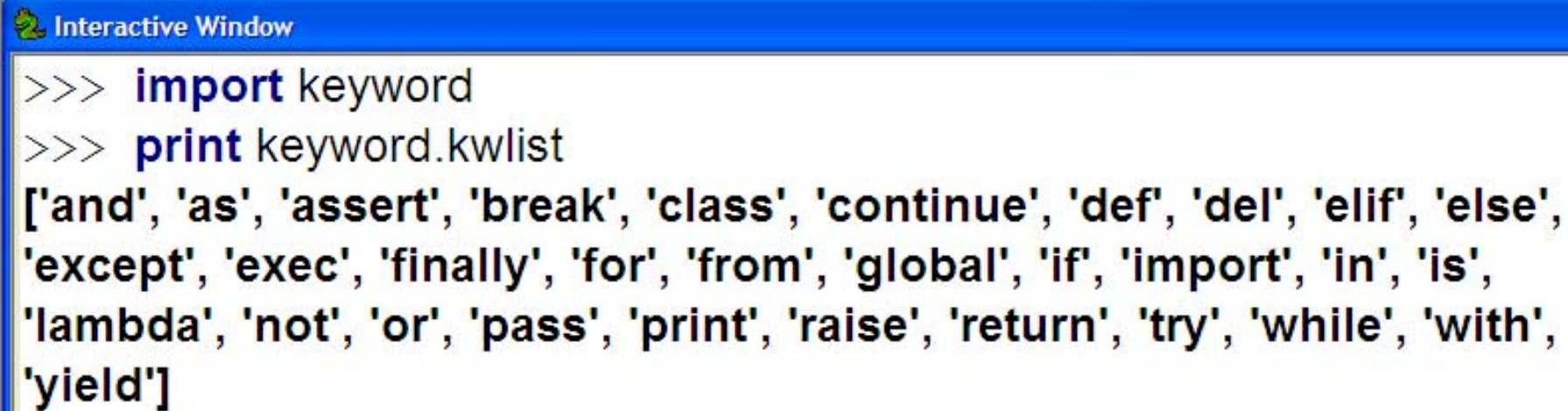
```
>>> from arcpy.mapping import *
>>> MapDocument(r'c:\temp\map_book.mxd')
<MapDocument object at 0xf07690[0xccfe3e0]>
```

Python Keywords



The screenshot shows the ArcMap interface with the Python console open. The console title is "Untitled - ArcMap - ArcInfo". The menu bar includes File, Edit, View, Bookmarks, Insert, Selection, Geoprocessing, Customize, and Windows. The toolbar shows various icons for file operations, navigation, and geoprocessing. The Python console displays the following code and error message:

```
>>> class = "Black Spruce Woodland"
Parsing error <type 'exceptions.SyntaxError':
```



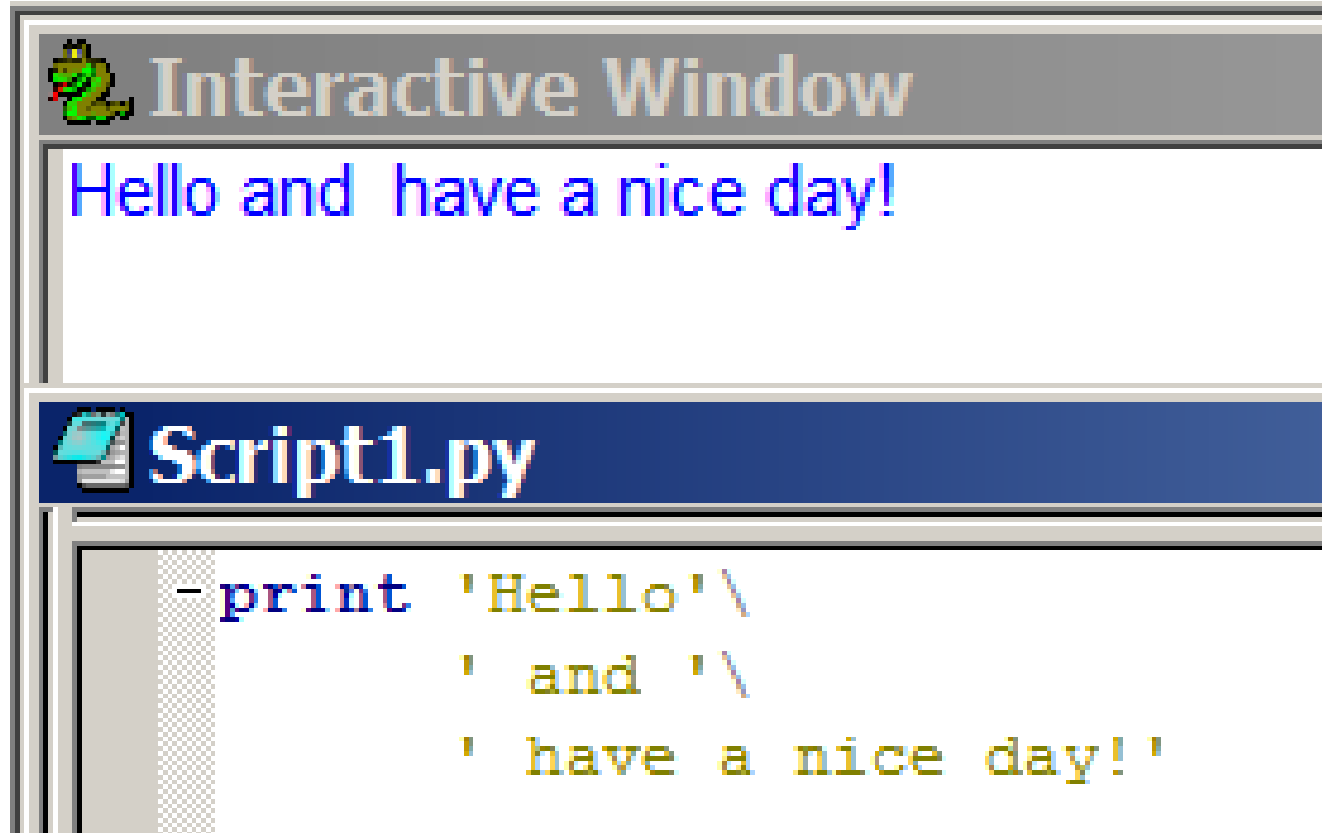
The screenshot shows the Interactive Window in ArcMap. The title bar is "Interactive Window". The console displays the following code and output:

```
>>> import keyword
>>> print keyword.kwlist
['and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else',
'except', 'exec', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',
'lambda', 'not', 'or', 'pass', 'print', 'raise', 'return', 'try', 'while', 'with',
'yield']
```




“\” character

\ is a special (\t means tab, \n means new line, etc.)
(use / or \\ for paths instead of \)



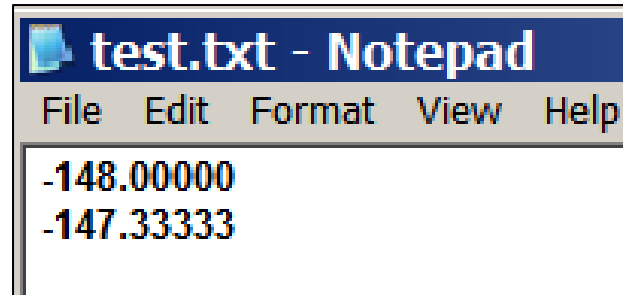


```
>>> badpath = 'c:\temp'
>>> goodpath1 = 'c:/temp'
>>> goodpath2 = 'c:\\temp'
>>> goodpath3 = r'c:\temp'
>>> print goodpath1,goodpath2,goodpath3
c:/temp c:\temp c:\temp
>>> os.chdir(goodpath1)
>>> os.chdir(goodpath2)
>>> os.chdir(goodpath3)
>>> print badpath
c:  emp
>>> os.chdir(badpath)
Traceback (most recent call last):
  File "<interactive input>", line 1,
WindowsError: [Error 123] The filename
label syntax is incorrect: 'c:\temp'
```

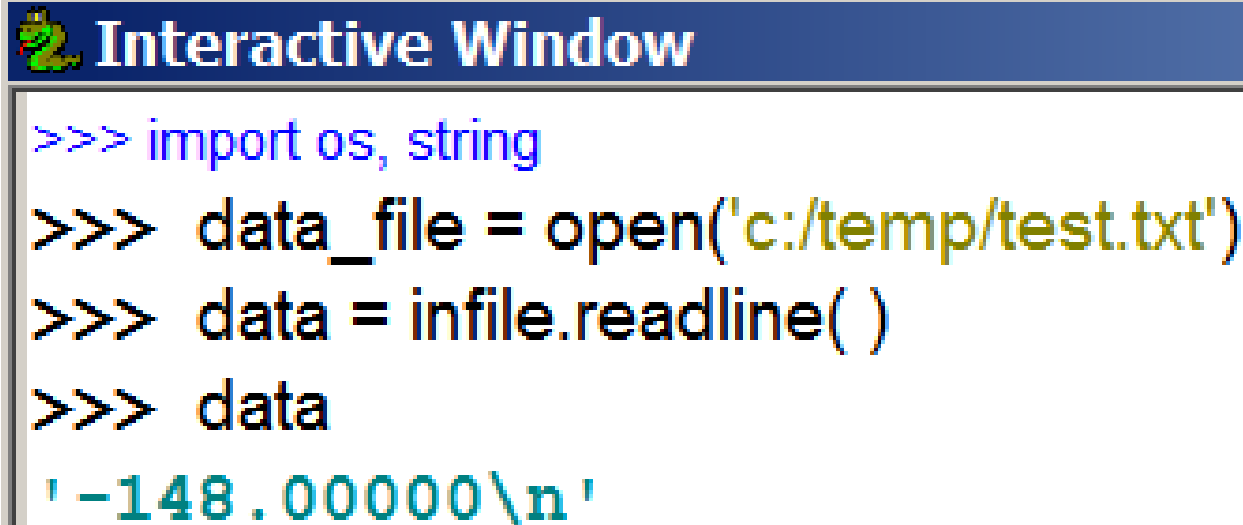


Newline = “\n”

- Newline character at the end of every line in a text file



```
test.txt - Notepad
File Edit Format View Help
-148.00000
-147.33333
```



```
Interactive Window
>>> import os, string
>>> data_file = open('c:/temp/test.txt')
>>> data = infile.readline( )
>>> data
'-148.00000\n'
```



Python Lists

```
>>> myList = [ 2011, 'ASMC', 64.75]
>>> myList[1]
'ASMC'
```

```
>>> myList[0]
2011
>>> myList[0:]
[2011, 'ASMC', 64.75]
```



Looping using range()

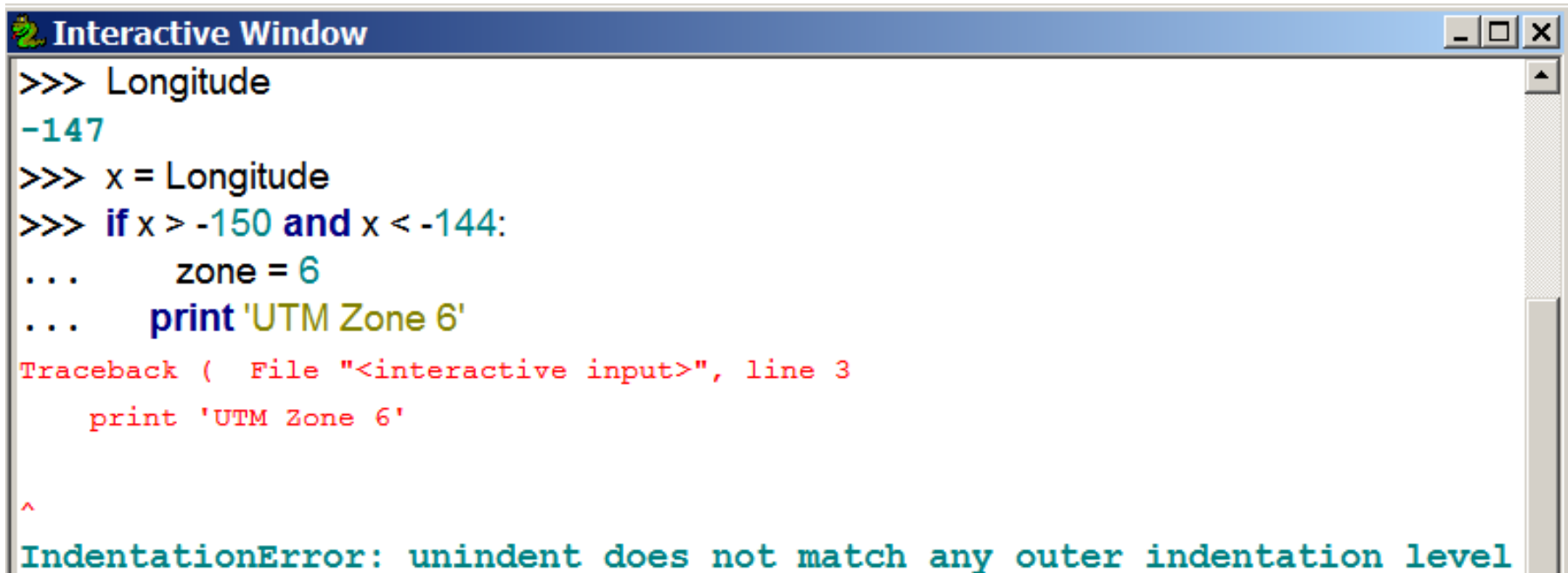
```
>>> myList
[2011, 'ASMC', 64.75]
>>> for item in range(0,2): #want list items 0,1,2
...     print myList[item]
...
2011
ASMC
```

```
>>> for item in range(0,3):
...     print myList[item]
...
2011
ASMC
64.75
```



: then indents

- Indentation interpreted as loop or decision structure



```
>>> Longitude
-147
>>> x = Longitude
>>> if x > -150 and x < -144:
...     zone = 6
...     print 'UTM Zone 6'
Traceback ( File "<interactive input>", line 3
      print 'UTM Zone 6'
      ^
IndentationError: unindent does not match any outer indentation level
```



Recycling Variables

- Some Python methods returns result None

```
>>> print strBuffer
100,meters
>>> strBuffer = strBuffer.replace(",", " ")
>>> print strBuffer
100 meters
```

```
>>> fList = ['point','polyline','multipoint','polygon']
>>> print fList
['point', 'polyline', 'multipoint', 'polygon']
>>> fList = fList.sort()
>>> print fList
None
```



Arcpy Site Package

- Python scripting language
- Arcpy geoprocessing

```
>>> import arcpy #arcpy site package
>>> arcpy.
```

Buffer_analysis
Buffer_arc
BuildBoundary_management
BuildFootprints_management
BuildNetwork_na



```
>>> lstFields = [ "field1","fieldb","fieldz"]
>>> arcpy.DeleteField_management("SALMON",lstFields)
<Result 'SALMON'>
```

```
Start Time: Tue Feb 15 14:36:11 2011
Dropping field1 from SALMON...
Dropping fieldb from SALMON...
Dropping fieldz from SALMON...
```

```
>>> strFields = ' field1;fieldb;fieldz '
>>> arcpy.DeleteField_management("SALMON",strFields)
<Result 'SALMON'>
```

```
Start Time: Tue Feb 15
Dropping field1 from S
Dropping fieldb from S
Dropping fieldz from S
```

```
>>> vt = arcpy.ValueTable()
>>> vt.addRow("field1")
>>> vt.addRow("fieldb")
>>> vt.addRow("fieldz")
>>> print vt
field1;fieldb;fieldz
```

```
>>> arcpy.DeleteField_management("SALMON",vt)
<Result 'SALMON'>
>>>
```

```
Start Time: Tue Feb 15 14:45:07 2011
Dropping field1 from SALMON...
Dropping fieldb from SALMON...
Dropping fieldz from SALMON...
```

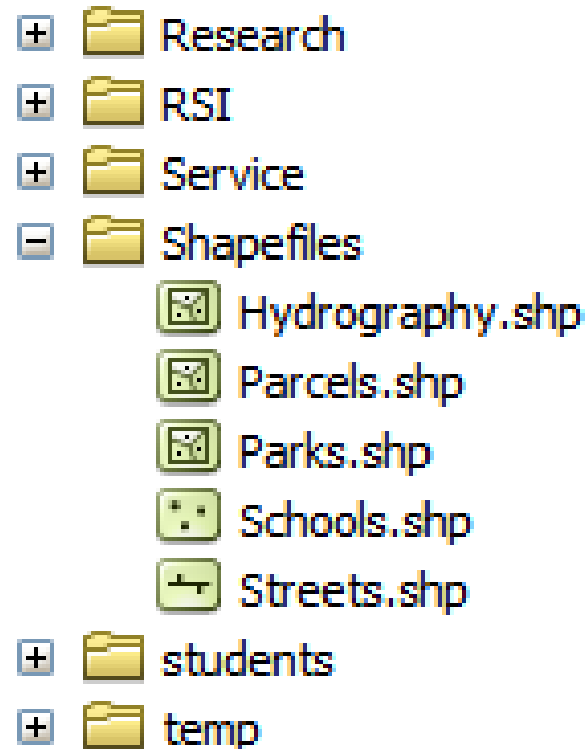
htt



```
>>> arcpy.Buffer_analysis(hydroFC,r'c:\temp\Buffer_Hydro100m.shp','100 METERS')
<Result 'c:\\temp\\Buffer_Hydro100m.shp'>
>>> print arcpy.GetMessages()
Executing: Buffer C:\Shapefiles\Hydrography.shp c:\temp\Buffer_Hydro100m.shp
"100 Meters" FULL ROUND NONE #
Start Time: Wed Jan 12 12:35:56 2011
Succeeded at Wed Jan 12 12:36:10 2011 (Elapsed Time: 14.00 seconds)
```

```
>>> arcpy.Buffer_analysis(hydroFC,r'c:\temp\BufferHydro100m.shp','100 meters')
<Result 'c:\\temp\\BufferHydro100m.shp'>
>>> print arcpy.GetMessages()
Executing: Buffer C:\Shapefiles\Hydrography.shp c:\temp\BufferHydro100m.shp
"100 Meters" FULL ROUND NONE #
Start Time: Wed Jan 12 12:33:11 2011
Succeeded at Wed Jan 12 12:33:25 2011 (Elapsed Time: 14.00 seconds)
```

```
>>> arcpy.buffer_analysis(hydroFC,r'c:\temp\Buffer_Hydro_100m.shp','100 METERS')
Traceback (most recent call last):
  File "<interactive input>", line 1, in <module>
AttributeError: 'module' object has no attribute 'buffer_analysis'
```



```
>>> import arcpy,glob,os
>>> myPath = r'C:\Shapefiles'
>>> os.curdir = myPath
>>> shapeFiles = glob.glob('*.shp')
>>> shapeFiles
```

```
['Hydrography.shp', 'Parcels.shp', 'Parks.shp', 'Schools.shp', 'Streets.shp']
```

```
>>> arcShapeFiles = arcpy.ListFeatureClasses()
```

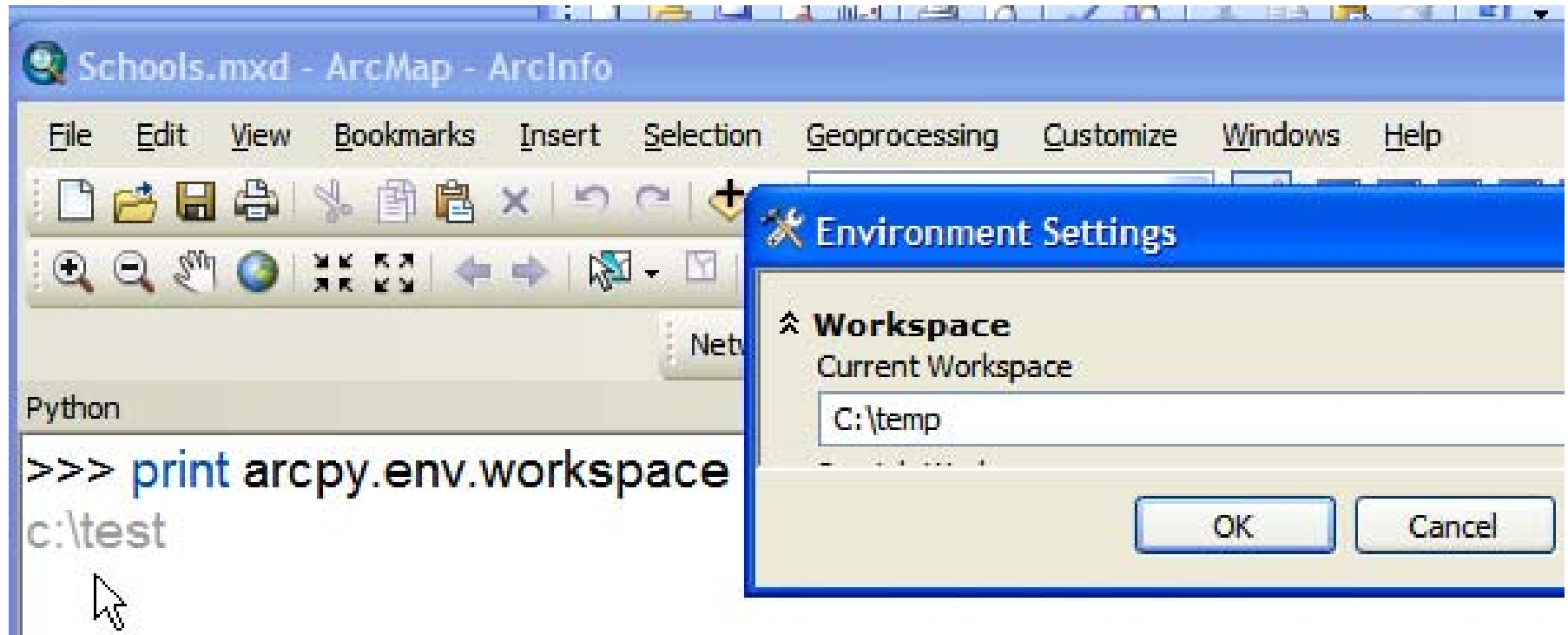
```
Traceback (most recent call last):
```

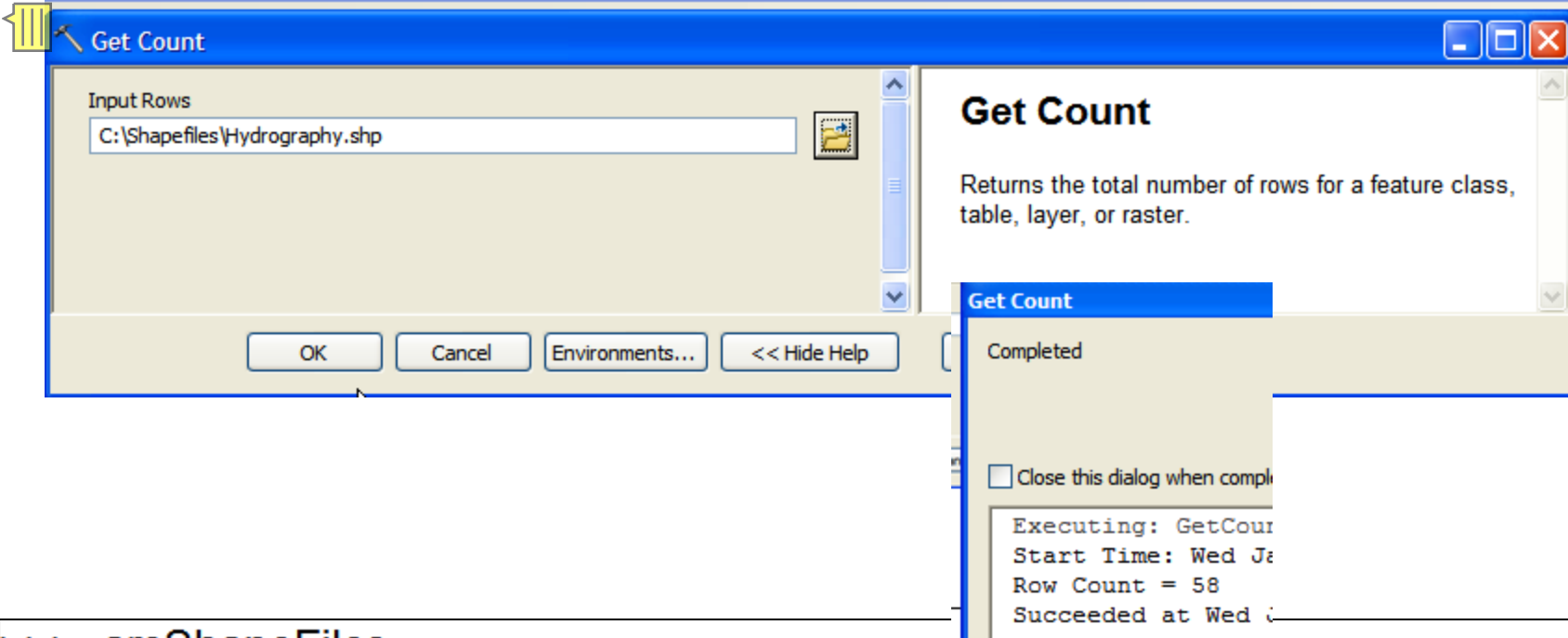
```
File "<interactive input>", line 1, in <module>
```

```
File "C:\Program
```

```
Files\ArcGIS\Desktop10.0\arcpy\arcpy\__init__.py", line 634, in
```

```
ListFeatureClasses
```





```
>>> arcShapeFiles
[u'Hydrography.shp', u'Parcels.shp', u'Parks.shp', u'Schools.shp', u'Streets.shp']
>>> hydroFC = arcShapeFiles[0] #first item in list
>>> nRecs = arcpy.GetCount_management(hydroFC) #get number of records
>>> print nRecs
58
>>> type(nRecs)
<class 'arcpy.arcobjects.arcobjects.Result'>
>>> lineCount = int(nRecs)
Traceback (most recent call last):
  File "<interactive input>", line 1, in <module>
TypeError: int() argument must be a string or a number, not 'Result'
```



```
>>> rows = arcpy.SearchCursor(hydroFC)
>>> sumArea = 0
>>> for row in rows:
...     sumArea = sumArea + row.shape.area
...
>>> print sumArea
279703875.005
```



Hydrography.shp.VERBYLA.12224.10900.sr.lock



Hydrography.shp.VERBYLA.12224.rd.lock

```
...
>>> print sumArea
279703875.005
>>> del rows,row #unlock feature class
```

Field Calculator

Parser: ☒ VB Script ☐ Python

Fields: OBJECTID, Shape, TYPE, Shape_Length, Root_Length

Type: ☒ Number ☐ String ☐ Date

Functions: Abs (), Atn (), Cos (), Exp (), Fix (), Int (), Log (), Sin (), Sqr (), Tan ()

☒ Show Codeblock

Pre-Logic Script Code:

```
dim sqr_Root
sqr_Root = Sqr ( [Shape_Length] )
```

Root_Length =

sqr_Root

Clear Load... Save... Help

OK Cancel

Field Calculator

Parser: ☐ VB Script ☒ Python

Fields: OBJECTID, Shape, TYPE, Shape_Length, Root_Length

Type: ☒ Number ☐ String ☐ Date

Functions: math.log10(), math.log1p(), math.modf(), math.pi(), math.pow(), math.radians(), math.sin(), math.sinh(), math.sqrt(), math.tan(), math.tanh(), math.trunc()

☒ Show Codeblock

Pre-Logic Script Code:

```
sqr_Root = math.sqrt( !Shape_Length!)
```

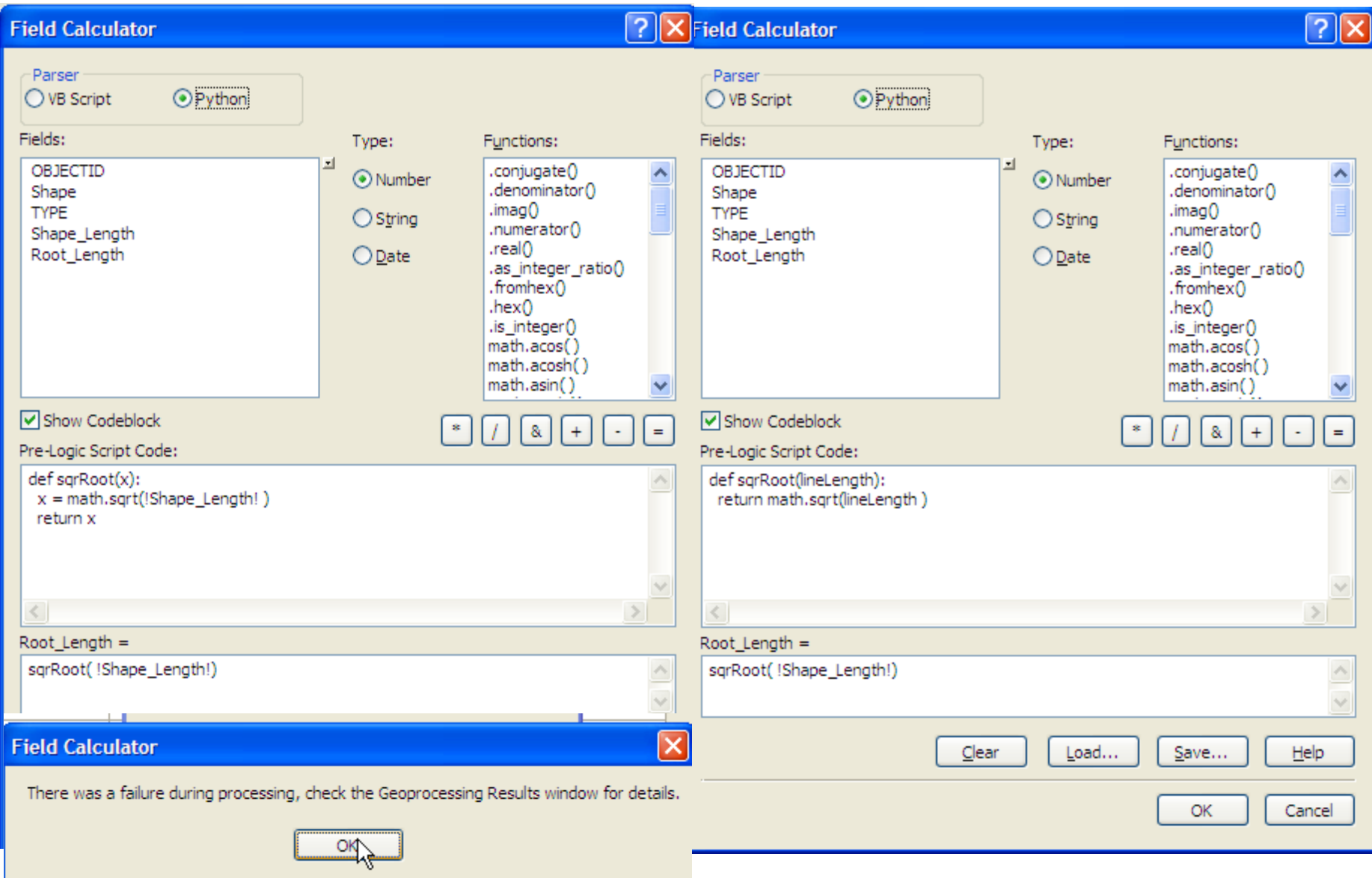
Root_Length =

sqr_Root

Field Calculator

There was a failure during processing, check the Geoprocessing Results window for details.

OK





Python

```
>>> arcpy.CalculateField_management(
```

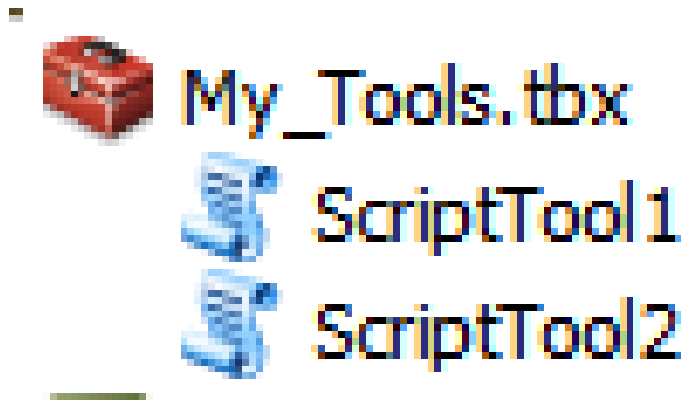
▢ "Points"

CalculateField_management(**in_table**, field, expression,
{expression_type}, {code_block})
Calculates the values of a field for a feature class, feature layer,
or raster
catalog.

INPUTS:
in_table (Table View or Raster Layer or Raster Catalog Layer or
Mosaic Layer):



Script Tools



- Built-in Dialogs
- Filtering to prevent errors
- Output to Arcmap Data Frame
- Portable (email the .tbx file and script)
- Toolbox, toolbar, or context menus

```
testParameters.py
import arcpy
x1 = arcpy.GetParameterAsText(0)
x2 = arcpy.GetParameterAsText(1)
x3 = arcpy.GetParameterAsText(2)
```

Add Script

Display Name	Data Type
Feature Layer	Feature Layer
Feature Class	Feature Class
Feature Dataset	Feature Dataset

Test Drive Script Tool Parameters

Feature Layer

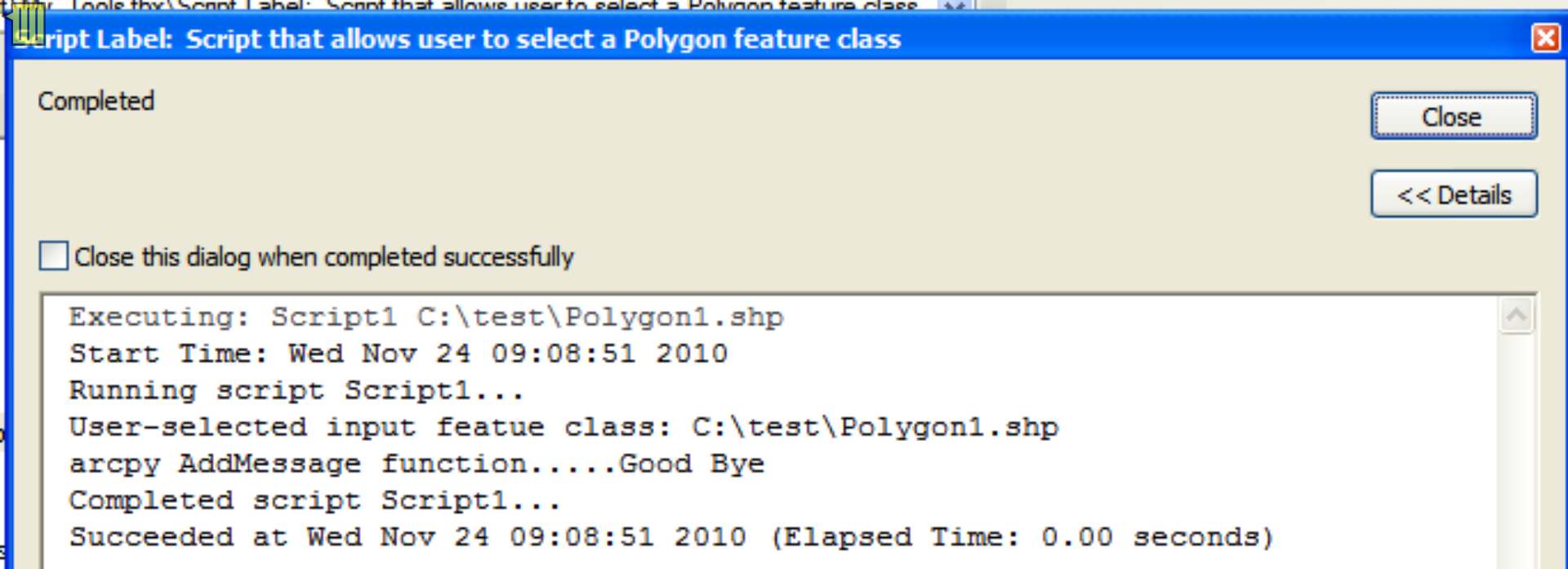
VegClasses

Feature Class

C:\nrm638\lab5_script_tools2\error_matrix\VegClasses.shp

Feature Dataset

C:\nrm435\verbyla\lines_lab.mdb\BearsFeatureDataset



```

#Python script to message to user
import arcpy
Theme = arcpy.GetParameterAsText(0) #user selects a feature class
#output using arcpy AddMessage function:
Message = 'User-selected input featue class: ' + str(Theme)
arcpy.AddMessage(Message)
arcpy.AddMessage('arcpy AddMessage function.....Good Bye')
#output using Python print command:
print 'Python print command...All Done'

```



Keep results younger than: 2 weeks

Display / Temporary Data

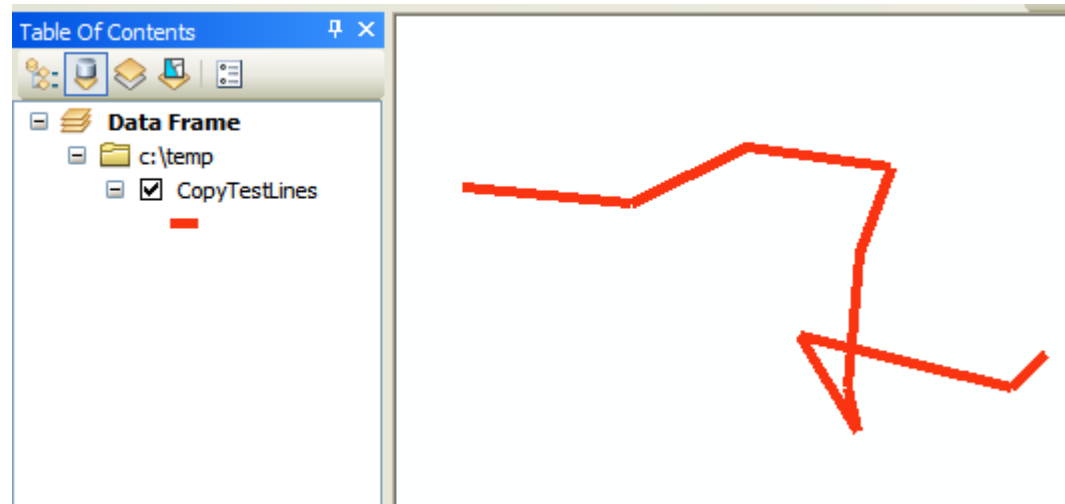
☒ Add results of geoprocessing operations to the display

☐ Results are temporary by default

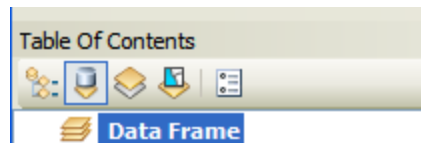
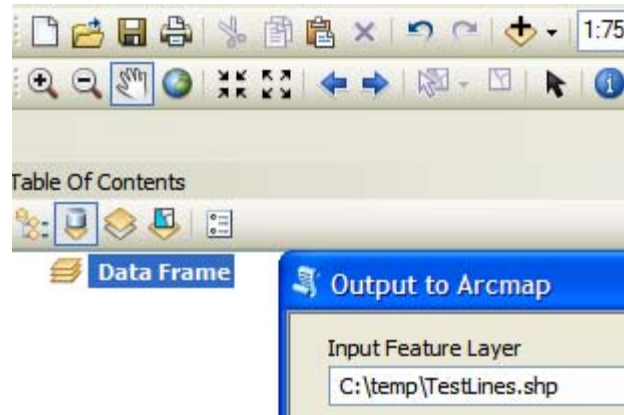
OK



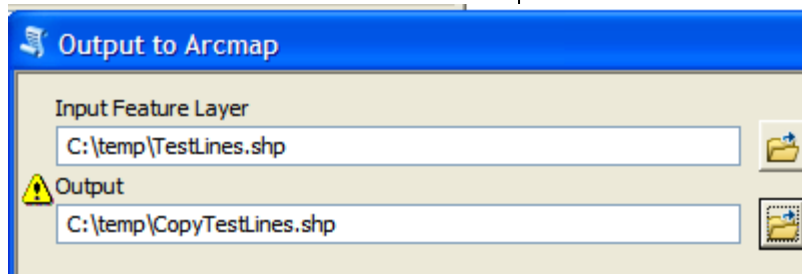
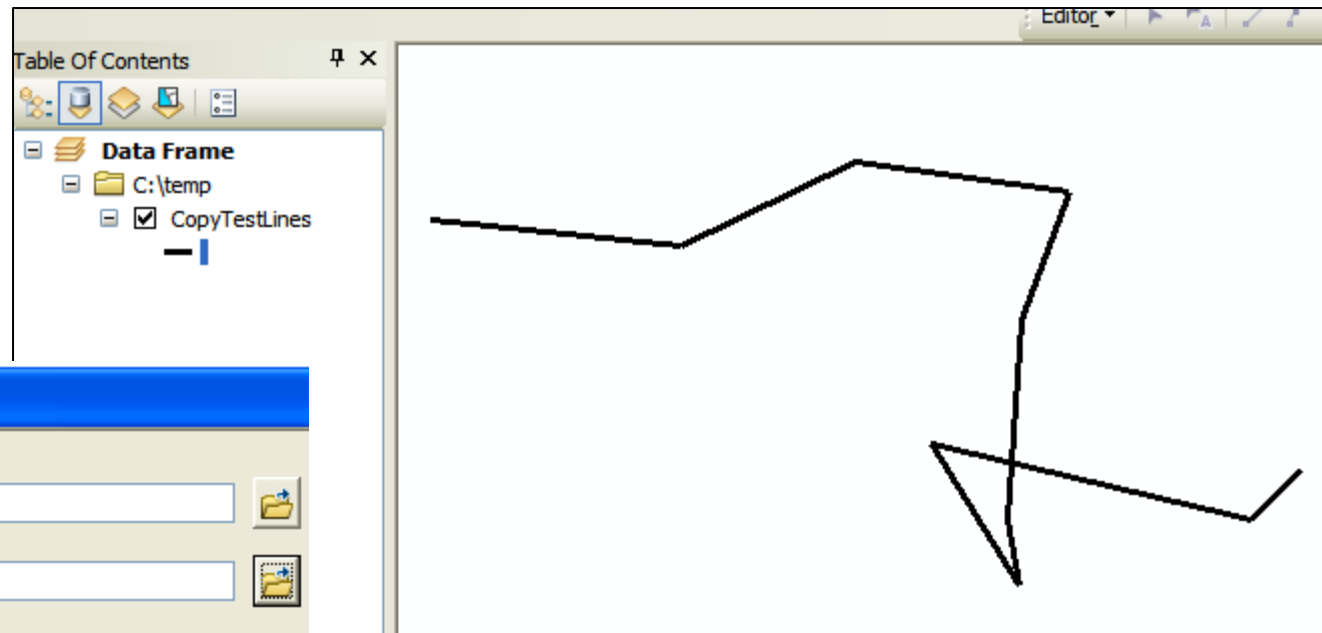
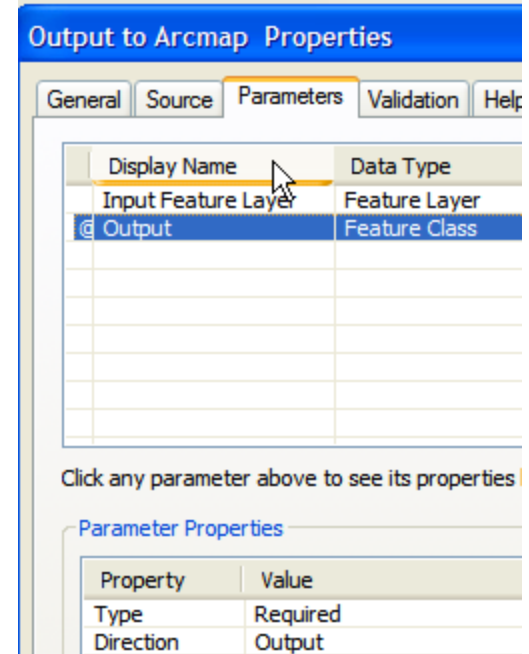
```
>>> import arcpy
... arcpy.env.workspace = r'c:\temp'
... inFC = 'TestLines.shp'
... outFC = r'CopyTestLines.shp'
... arcpy.CopyFeatures_management(inFC,outFC)
```



```
import arcpy
arcpy.env.workspace = r'c:\temp'
inFC = arcpy.GetParameterAsText(0)
outFC = r'CopyTestLines.shp'
arcpy.CopyFeatures_management(inFC,outFC)
```



```
import arcpy
arcpy.env.workspace = r'c:\temp'
inFC = arcpy.GetParameterAsText(0)
#outFC = r'CopyTestLines.shp'
outFC = arcpy.GetParameterAsText(1)
arcpy.CopyFeatures_management(inFC,outFC)
```





Points				
	FID	Shape	ptDate	Days
	0	Point	12/31/2010	0
	1	Point	1/10/2011	0
	2	Point	1/16/2011	0
	3	Point	1/22/2011	0
	4	Point	1/25/2011	0
	5	Point	1/26/2011	0
	6	Point	1/30/2011	0
	7	Point	2/1/2011	0
	8	Point	2/2/2011	0
	9	Point	2/4/2011	0

Point Movement

Animal Locations Points Theme:

Points

Field containing point date:

|

Point Movement Properties

General Source Parameters Validation Help

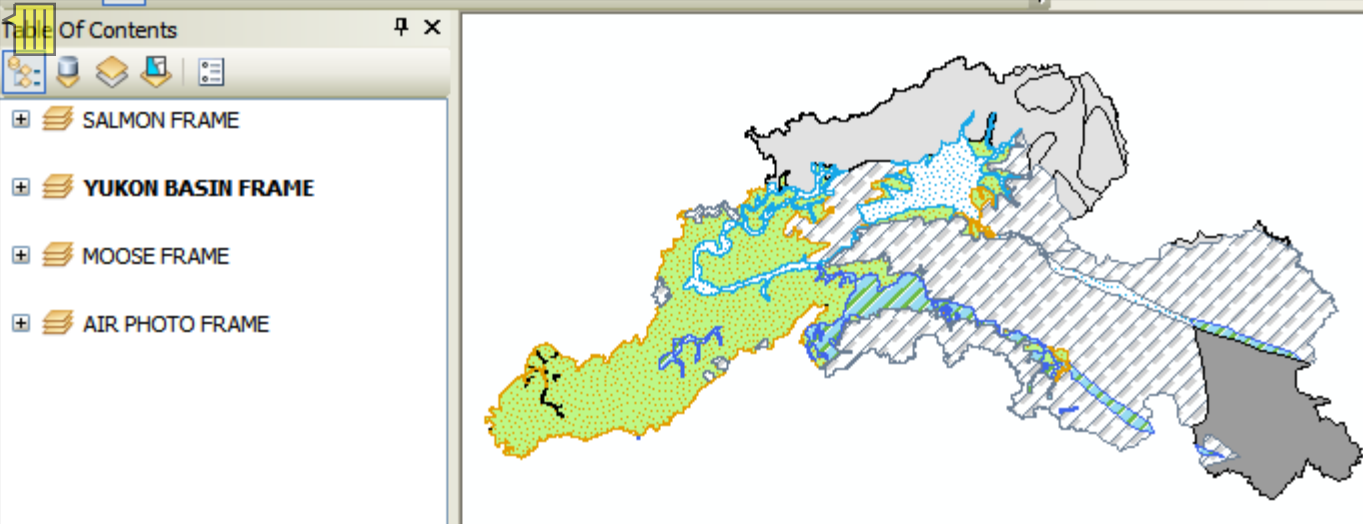
Display Name	Data Type
Animal Locations Poi...	Feature Layer
@ Field containing point ...	Field

Parameter Properties

Property	Value
Type	Required
Direction	Input
MultiValue	No
Default	
Environment	
Filter	Field
Obtained from	Animal_Locations_Points_Theme_



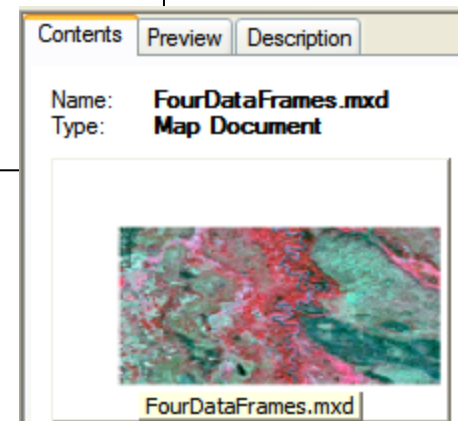
```
calcDist = """Record=0
def PointDistance(x,y):
    #compute distance traveled between 2 points
    global oldX,oldY,Record
    if(Record==0):
        Dist=0.0
    else:
        Dist=math.sqrt((x - oldX)**2+(y - oldY)**2)
    oldX = x
    oldY = y
    Record=Record+1
    return Dist"""
expression = "PointDistance(!POINT_X!,!POINT_Y!)"
#now use Calculate Field tool to compute distance between points
arcpy.CalculateField_management(inPoints ,"Distance",expression,"PYTHON",calcDist)
```

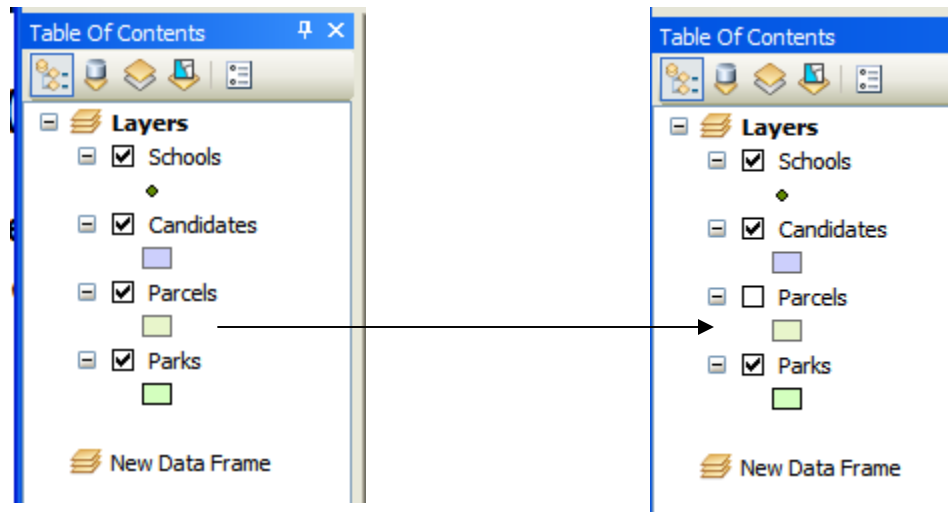


make_mxd_thumbnail.py

```
import arcpy
mxdFile = r'C:\temp\FourDataFrames.mxd'
mapDoc = arcpy.mapping.MapDocument(mxdFile)
framesList = arcpy.mapping.ListDataFrames(mapDoc)
mapDoc.activeView = framesList[3] #fourth data frame
mapDoc.makeThumbnail()
mapDoc.save()
```

<http://nrm.salrm.uaf.edu/~dverbyla>





```
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:59,064
Network Analyst Network Dataset:
Python
>>> mapDoc = arcpy.mapping.MapDocument("Current")
>>> dataFrame = arcpy.mapping.ListDataFrames(mapDoc) [0]
>>> parcelsLayer = arcpy.mapping.ListLayers(mapDoc,"Parcels*",dataFrame) [0]
>>> parcelsLayer.visible = False
>>> arcpy.RefreshTOC()
>>> arcpy.RefreshActiveView()
```



The screenshot displays the ArcGIS Desktop interface. On the left, the 'Contents' pane shows a map document named 'polygon_analysis_lab.mdb' containing several layers: 'city_parcels', 'Grizzly_Bears', 'mining_claim', 'veg_polys', and 'Schools'. The 'Schools' layer is highlighted. In the top right, a status bar shows coordinates: Left: 1847318.862754 ft, Right: 1859639. ft, and Bottom: 765532.642027 ft. Below this, the 'Data Source' pane shows the following information:

Data Type:	Personal Geodatabase Feature Class
Feature Class:	Schools
Location:	C:\temp\polygon_analysis_lab.mdb
Feature Dataset:	city_parcels
Feature Type:	Simple

Below the 'Data Source' pane, the 'Table Of Contents' pane shows the 'Layers' list with 'C:\temp\polygon_analysis_lab.mdb' and 'Schools' (checked). At the bottom, the 'Python' console shows the following code:

```
>>> mapDoc = arcpy.mapping.MapDocument("current")
>>> brokeLayer = arcpy.mapping.ListBrokenDataSources(mapDoc)[0]
>>> brokeLayer
<map layer u'Schools'>
>>> brokeLayer.replaceDataSource(r'C:\temp\polygon_analysis_lab.mdb')
>>> arcpy.RefreshTOC()
```



```
Python
>>> mapDoc = arcpy.mapping.MapDocument("Current")
>>> brokeLayer = arcpy.mapping.ListBrokenDataSources(mapDoc) [0]
>>> brokeLayer
<map layer u'Schools'>
>>> brokeLayer.replaceDataSource(r"C:\temp","SHAPEFILE_WORKSPACE","Schools.shp")
Runtime error <type 'exceptions.ValueError': Layer: Unexpected error
>>>
```

Parameter	Explanation	Data Type
workspace_path	A string that includes the workspace path to the new data or connection file.	String
dataset_name	A string that represents the name of the dataset the way it appears in the new workspace (not the name of the layer in the TOC).	String

- SHAPEFILE_WORKSPACE
—A shapefile workspace



Beginner Websites(hyperlinks)

- [Non-Programmer's Tutorial for Python 2.6](#)
- [Live Wires Python Course](#)
- [Instant Hacking](#)
- [Dive Into Python](#)
- [Python Programming for Beginners](#)
- [Python 101](#)
- [Learn Python in 10 Minutes](#)
- [Learn Python](#)
- [5 Minutes With Python \(video\)](#)
- [Online Python resources \(video\)](#)



ESRI Resources

Using Python in ArcGIS10

Free Web Course <http://training.esri.com>

- Creating basic Python scripts with correct syntax
- Arcmap Python window
- Python scripting in field calculator
- Creating basic script tools in ArcGIS10

Getting Started With Python in ArcGIS10

Video <http://resources.arcgis.com/gallery/video/geoprocessing/>

- Arcpy site package
- Python window
- Script tools
- Arcpy mapping automation
- Tool design and validation



ESRI Resource Center

<http://resources.arcgis.com/gallery/file/geoprocessing/>

UC 2010 Tech Workshop: Python Essentials in ArcGIS I



By *drewf*

★★★★★ (1 rating)



[ArcGIS.com](#)

[Support](#)



ArcGIS Resource Center

[Help](#)

[Blogs](#)

[Forums](#)

[Home](#) » [Geoprocessing](#) » [Geoprocessing Model and Script Tool Gallery](#)

UC 2010 Tech Workshop: Python Essentials in ArcGIS II



By *drewf*

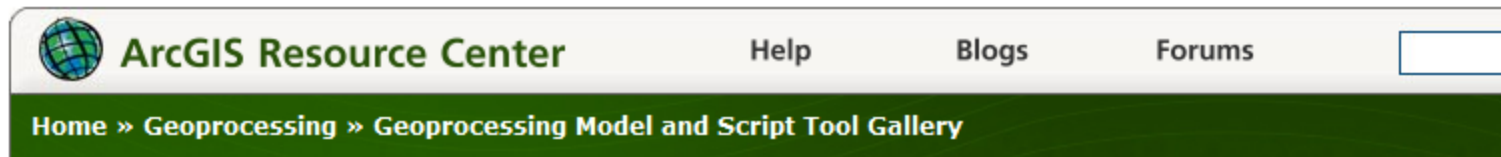
★★★★★ (1 rating)



<http://nrm.salrm.uaf.edu/~dverbyla>



ESRI Resource Center

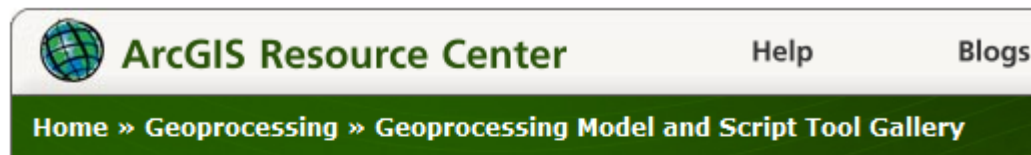


UC 2010 Technical Workshop: Building Geoprocessing tools

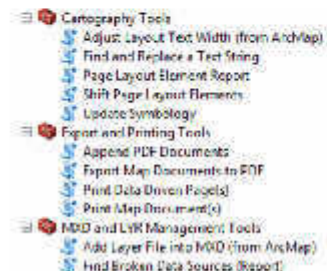


By *dmhoneycutt*

★★★★★ (0 rating)



arcpy.mapping sample script tools



By *jbarrette*

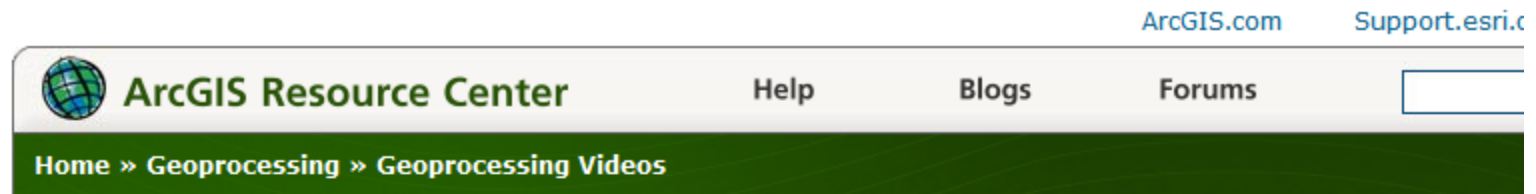
★★★★★ (6 ratings)



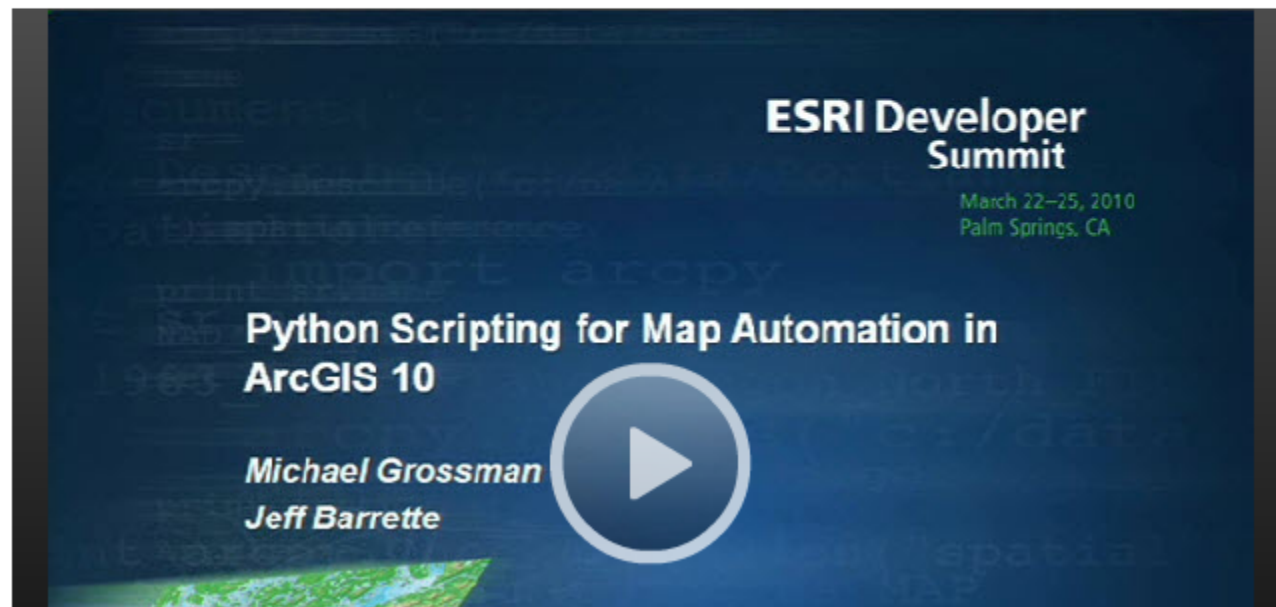


ESRI Resource Center

<http://resources.arcgis.com/gallery/video/geoprocessing>



DS2010: Python Scripting for Map Automation in ArcGIS 10



<http://nrm.salrm.uaf.edu/~dverbyla>