

# Python

**Christopher Grasteit**

# Agenda

- **Why write scripts**
- **What is Python and why use it**
- **Python Basics**
- **Using Python with ArcGIS**
- **Python Resources**



# Why write scripts

- Automate a work flow
- Run code at specific dates and times
- Easily distribute code

# What is Python and why use it

- **An open-source, object-oriented, scripting language**
  - Can view and modify source code
  - Support for large projects
  - Easy to use
- **Offer IDE's with debugging tools**
- **Modular, can be broken apart**
- **Cross Platform and works in many web-browsers.**
- **Installed with ArcGIS 9 and ESRI samples provided**

# Python Basics

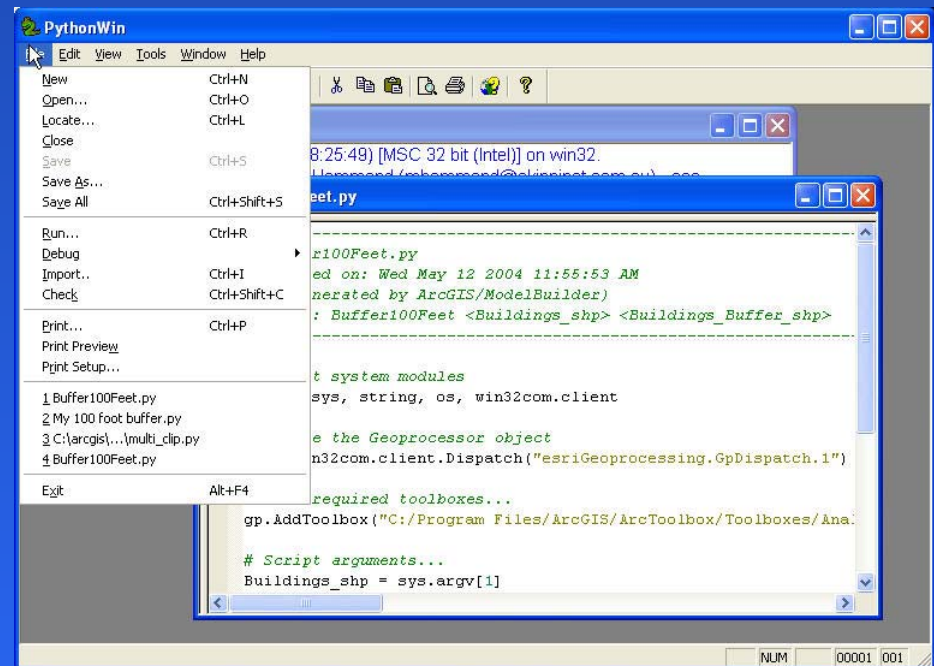
- PythonWin Interface
- IDLE Interface
- Comments
- Variables
- Strings, Numbers & Lists
- Built-in Functions
- Accessing modules
- Statements
- Decision making syntax, looping

# Where to Write Code

- **Many places to write Python code**
  - Text Editor: Notepad, Wordpad, etc...
  - Python command prompt
  - IDE: PythonWin, IDLE, etc...
- **IDE's allow you to perform all jobs from one location**
  - Write, save, run, and debug code from one location
- **Command prompt and both IDE's installed with ArcGIS**

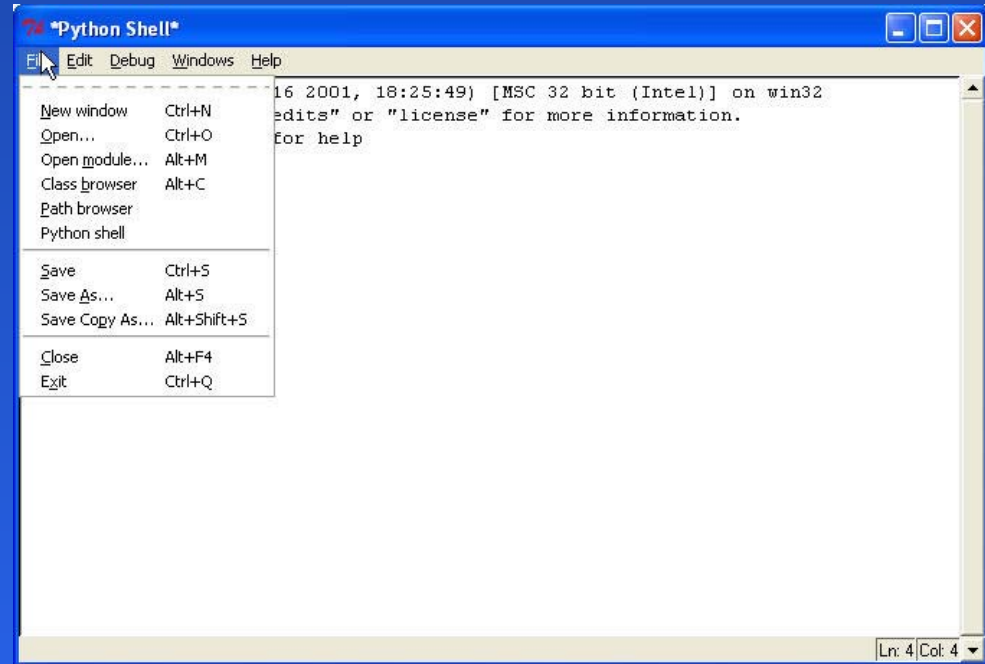
# PythonWin interface

- Menus, toolbars, and context menus
- Script window
  - Write and save code
- Interactive window
  - Test lines of code
  - Report messages
- Benefits
  - Windows look and feel
  - All in one application
  - Script tools open in PythonWin



# IDLE Interface

- **Menus**
- **Script window**
  - Write and save code
- **Python shell**
  - Test lines of code
  - Report messages
- **Benefits**
  - Cross-platform
  - Stable (e.g., continuous loops)



# Comments

- **Comment: A non-executable line of code**
  - One number sign (#) for green and italicized
  - Two number signs (##) for grey

```
# -----  
## Buffer100Feet.py  
## Created on: Wed May 12 2004 11:55:53 AM  
## (generated by ArcGIS/ModelBuilder)  
## Usage: Buffer100Feet <Buildings_shp> <Buildings_Buffer_shp>  
# -----  
  
# Import system modules  
import sys, string, os, win32com.client  
  
# Create the Geoprocessor object  
gp = win32com.client.Dispatch("esriGeoprocessing.GpDispatch.1")  
  
# Load required toolboxes...  
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Analysis Tools.tbx")  
  
# Script arguments...  
Buildings_shp = sys.argv[1]  
if Buildings_shp == '#':  
    Buildings_shp = "C:/Data/SanFranciscoZoo/Cleanup Hold/Data/Buildings.shp" # provide a default value if unspecified  
  
Buildings_Buffer_shp = sys.argv[2]  
if Buildings_Buffer_shp == '#':  
    Buildings_Buffer_shp = "C:/Data/SanFranciscoZoo/Cleanup Hold/Data/Buildings_Buffer.shp" # provide a default value if unspecified  
  
# Local variables...  
  
# Process: Buffer...  
gp.Buffer_analysis(Buildings_shp, Buildings_Buffer_shp, "100.000000 Feet", "FULL", "ROUND", "NONE", "")
```

# Variables in Python

- **Variables are dynamically typed**
  - No declaration keyword
  - No type assignment

```
Fc = "c:/data/test.shp"
```
- **Variables are case sensitive**

```
scale = 10000  
Scale = 20000
```
- **Variables can hold different data types**
  - Strings, numbers, lists, etc...

# Strings, Numbers and Lists

- **Variables can hold strings, numbers and lists**
  - Folder = "c:\data"
  - Num1 = 1.2
  - numList = [1, 2, 3]
- **Strings surrounded in double(") or single(') quotes**
- **Pathnames use two back(\\) or one forward(/) slash**
- **Strings can be combined together**
  - Gdbpath = "c:\\data"
  - Fc = "roads.shp"
  - myFile = Gdbpath + "\\ " + Fc

# Built-In Functions

- Python has many built-in functions
  - `len()` = *Returns the length*  
`Fc = "Railroads.shp"`  
`len(Fc)`
  - `max()` = *Returns the maximum value*  
`xExtent = (6260474.996464, 6338807.996464)`  
`max(xExtent)`
  - `open()` = *Opens a file*  
`Coord = open("c:\\xy.txt", "r").read()`
  - `round()` = *Rounds a number*  
`yCoord = 1811884.623964`  
`round(yCoord)`

# Accessing Modules

Most functions need to be imported from modules

- The math module

```
import math  
math.sqrt(64)
```

- The string module

```
import string  
string.upper("c:\\data")
```

- The os.path module

```
import os.path  
os.path.basename("c:\\data\\streets.shp")
```

# Statements

Python has many statements

- **print = Sends output to the interactive window**

```
print "Hello"
```

- **import = Imports a module**

```
import math
```

```
import string
```

- **Other Statements**

```
if...elif...else
```

```
while
```

```
for...in
```

```
try...except
```

# Decision Making Syntax

- **Testing conditions**

```
if x == 1:  
    print "x is 1"  
elif x == 2:  
    print "x is 2"  
else:  
    print "x is not 1 or 2"
```

- **Colons used at end of each condition**
- **Indentation defines what executes for each condition**
  - Python automatically indents when you press Enter
  - Use tabs or spaces, must be consistent
- **Two equal signs for conditions, one for assignment**

# Looping Syntax

- **While loops, counted loops and list loops**

```
x = 5
```

```
while x < 1:
```

```
    print x
```

```
    x = x + 1
```

```
for x in range (1,5):
```

```
    print x
```

- **Colons used at end of each statement**
- **Indentation defines what executes for the loop**

# Using Python with ArcGIS

- Python can be used to access the GeoProcessor
- Where you can use Python
  - Command Prompt
  - ArcToolbox
  - Model builder will export to Python
- Tell Python you want to use COM objects
  - Import win32com.client
- Create the Geoprocessor
  - Gp =  
win32com.client.Dispatch("esriGeoprocessing.GpDispatch.1")

# Python Resources

- **Books**
  - Learn to Program Using Python
  - Learning Python
  - The Quick Python Book
  - Python, Essential Reference
  - Python Programming on Win32
  - ... many more
- **Web site: [www.python.org](http://www.python.org)**
  - Tutorials, documentation, and forums

# Questions