>>> Python and GIS:
… Beyond ModelBuilder
… and PythonWin

Chad Cooper
Southwestern Energy Company

>>> What we'll cover

> Lists, lists, lists
> Code examples
> 3rd party modules
> Editors/IDEs
> Python references
> This presentation/code online!

Code in orange
Pseudocode in blue
Output/results in green

>>> Love the list

> List: Array of object references - Huh? *or* Ordered collection of arbitrary objects

```
list1 = [0,1,2,3]
list2 = ['zero','one','two','three']
list3 = [0,'zero',1,'one',2,'two',3,'three']
```

> Ordered

```
list2.sort()                    list2.sort(reverse=True)
['one','three',...]             ['zero','two',...]
```

> Mutable

```
list1.append(4)                 list1.reverse()
[0,1,2,3,4]                     [4,3,2,1,0]
```

>>> Keep lovin' the list

> Iterable `for l in list3`

```
0
zero ...
```

> Membership `3 in list3 --> True`

> Nestable - 2D array/matrix

```
list4 = [[0,1,2],
         [3,4,5],
         [6,7,8]]
```

> Access by index - zero-based

```
list4[1]      list4[1][2]
[3,4,5]       5
```

>>> Process a text file

```
 1 import csv, arcgisscripting
 2 gp = arcgisscripting.create()
 3 rows = gp.InsertCursor('C:/database.gdb/table')
 4 fields = ['slat','slon','spud']
 5 reader = csv.reader(open('C:/file.csv','rb'))
 6 for line in reader:
 7     t = 0
 8     row = rows.NewRow()
 9     for field in fields:
10         row.SetValue(field,line[t])
11         t = t + 1
12     rows.InsertRow(row)
```

CSV

1	37.50436,-91.93574,5/25/1986
2	35.25364,-93.48511,4/2/1997
3	34.69587,-90.59364,1/19/2005
4	...
5	...

GDB table

	slat	slon	spud
▶	37.50436	-91.93574	5/25/1986
	35.25364	-93.48511	4/2/1997
	34.69587	-90.59364	1/19/2005

>>> Databases:
MS SqlServer

```
1 import ceODBC
2 db = ceODBC.connect('DSN=Wells')
3 c = db.cursor()
4 c.execute('SELECT wellname, slat, slon \
5     FROM wells WHERE wellid IN (1,2,3)')
6 for each in c.fetchall():
7     print each

[('DOE #1-18', 33.095599, -92.38563),
 ('SMITH #2-7', 35.10155, -91.48824),
 ('JOHN #11-13', 34.09130, -93.45256)]
```

>>> Databases:
IBM DB2

```
1 import ceODBC
2 db = ceODBC.connect('DSN=dsn;UID=uid;PWD=pwd')
3 c = db.cursor()
4 c.execute('SELECT wellname FROM wells')
5 for each in c.fetchmany(3):
6     print each
```

```
[('DOE #1-18', 33.095599, -92.38563),
 ('SMITH #2-7', 35.10155, -91.48824),
 ('JOHN #11-13', 34.09130, -93.45256)]
```

>>> Sending email

```
1 import smtplib
2 server = smtplib.SMTP(email_server_ip)
3 msg = 'Eat more spam and eggs!'
4 server.sendmail('from@me.com',
5                 'to@you.com',
6                  msg)
7 server.quit()
```

> Make sure port 25 is unblocked!

>>> Download files

> GNU wget - free CLI program
> Retrieve using HTTP protocol
> Build a URL such as:
'http://www.archive.org/download/usgs_drg_ar_36094_a1/o36094a1.tif'
> Pass URL to wget, it fetches

>>> Download files

```
 1 import os, shutil
 2 wgetDir = 'C:/Program Files/wget/o'
 3 quads =
 4 [['36094a1','ELKINS','ar'],['36094a2','FAYETTEVILLE','ar']]
 5 exts = ['tif', 'tfw', 'fgd']
 6 url = 'http://www.archive.org/download/'
 7 home = 'C:/temp/quads/'
 8 for quad in quads:
 9     for ext in exts:
10         fullurl = url + 'usgs_drg_' + quad[2] + \
11         '_' + quad[0][:5] + '_' + quad[0][5:] + \
12         '/o' + quad[0] + '.' + ext
13         os.system('wget %s -o log.log' % (fullurl))
14         # Move and rename --> AR_ELKINS_o35094h3.tif
15         shutil.move(wgetDir + quad[0] + '.' + ext,
16             home + quad[2].upper() + '_' + \
17             quad[1].replace(' ','_') + \
18             '_' + 'o' + quad[0] + '.' + ext)
```

>>> DB records --> FC

```
 1 connect(dsn)
 2 db.cursor()
 3 execute(sql)
 4 results = fetchall()
 5 fields = [f1,f2,...]
 6 rows = InsertCursor(table)
 7 for r in results:
 8     row = rows.NewRow()
 9     row.SetValue(fields[i], r[i]
10     ...
11 if point data:
12     MakeXyEventLayer(table,x,y,layer)
13     CopyFeatures(layer,featureclass)
14 else:
15     AddJoin(layer,layerfield,table,tablefield)
16     CopyFeatures(layer,featureclass)
```

>>> 3rd party modules

> pyExcelerator, xlrd - Excel
> ceODBC, MySQLdb - databases
> mxDateTime - date/time f(x)s
> csv* - read/write csv files
> geopy - geocode using 3rd party geocoders (Google, Yahoo, VE)
> For fun: gdata (Google API), Flickr, digital photos (EXIF)

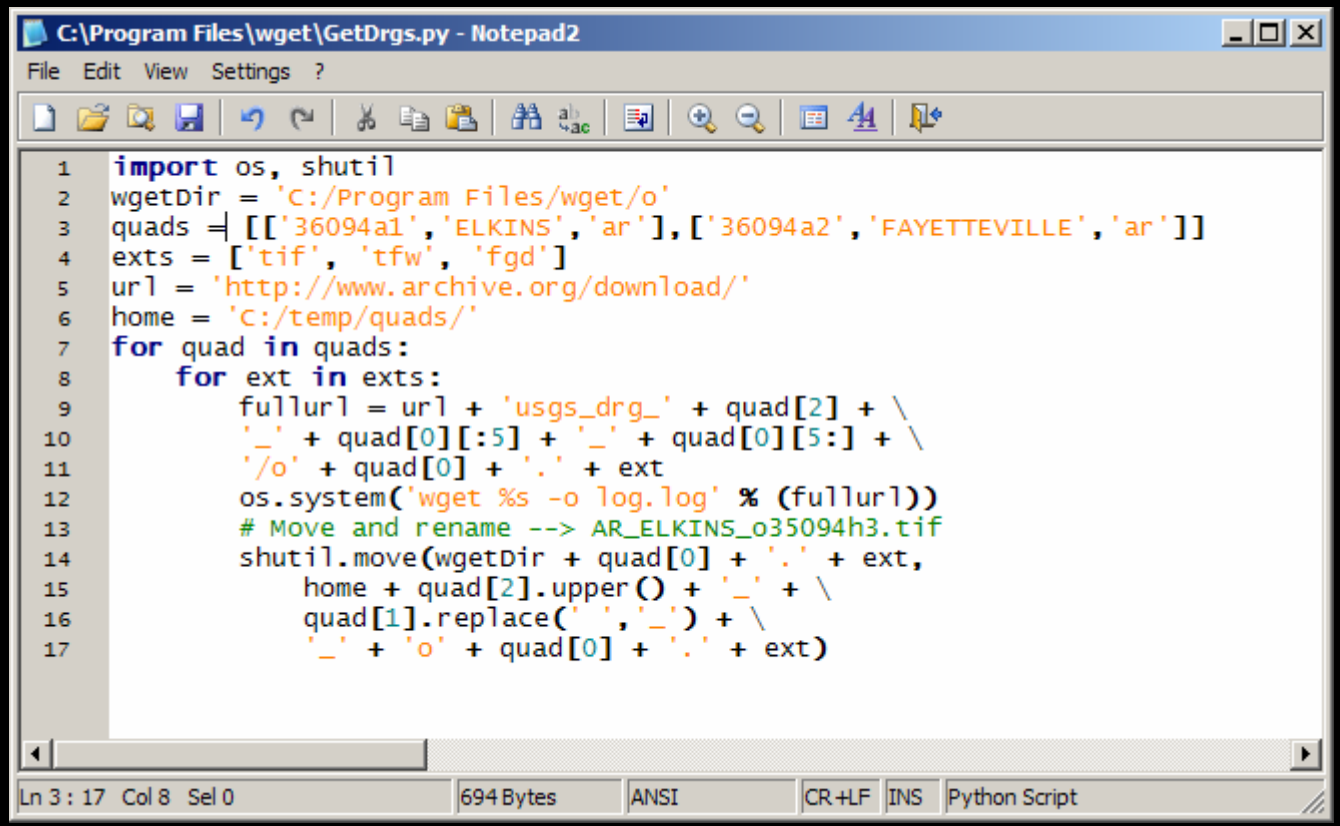
* in standard library

>>> Editors - Notepad

> Editor: Basic - you write code

>>> Editors - Notepad2

> Editor: Basic - you write code



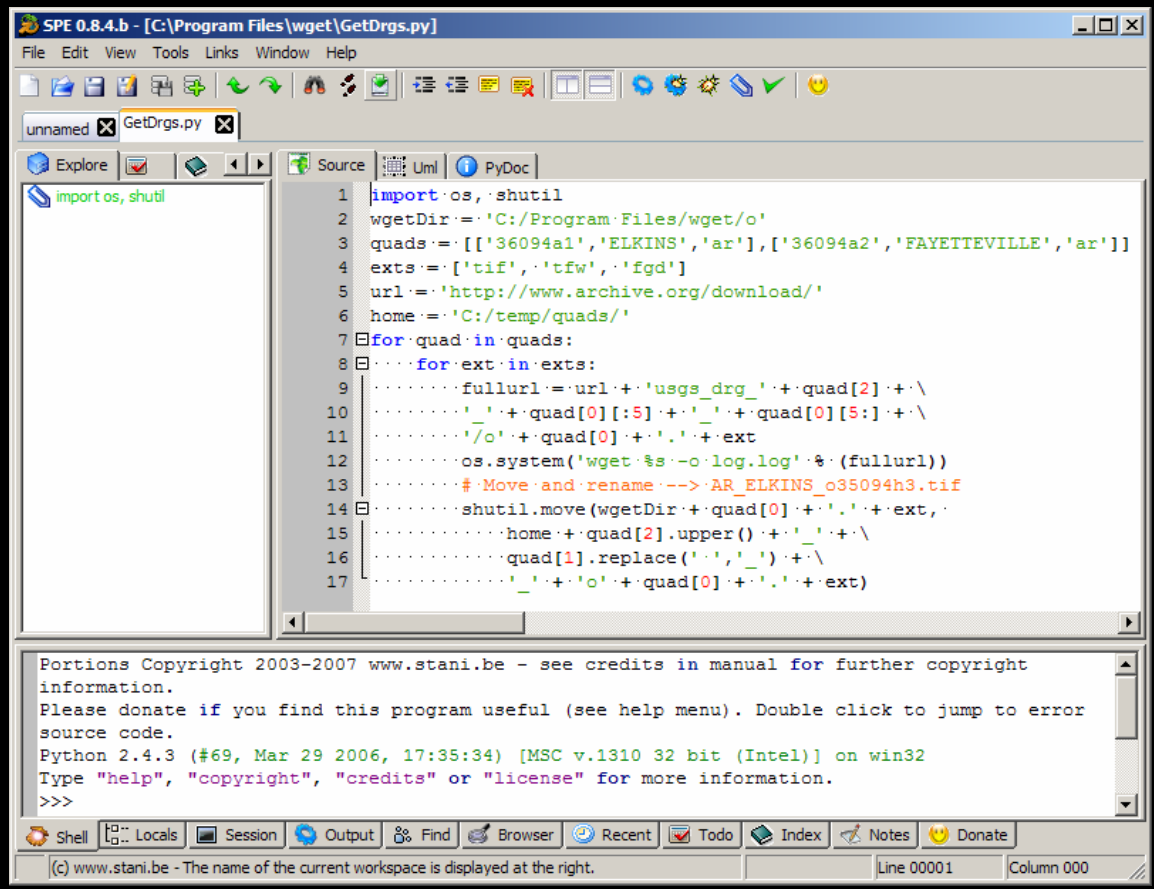
>>> IDEs - SPE

> IDE: Integrated development environment

> Tools

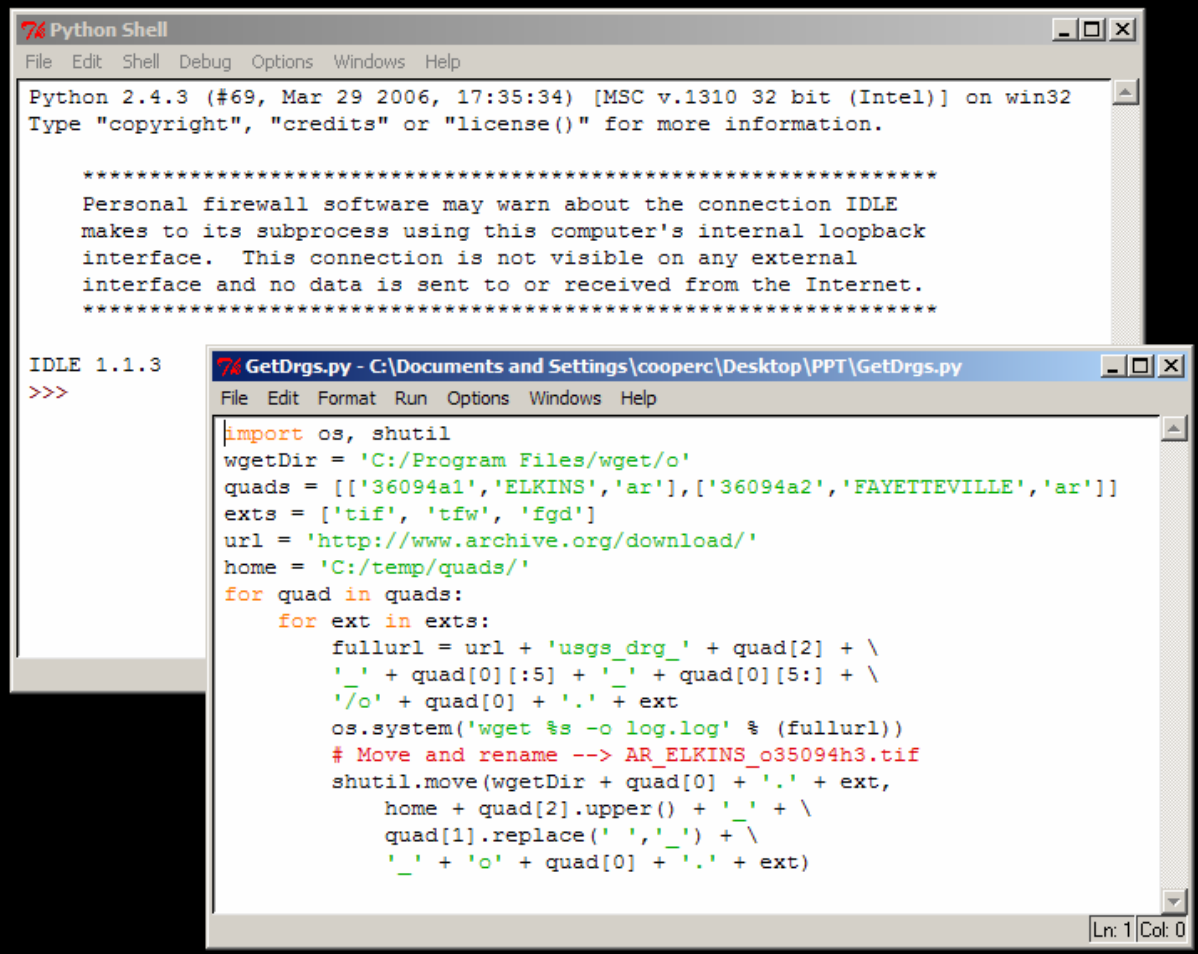
> Debugger

> Shell



>>> IDEs - IDLE

> Ships with CPython distro



>>> Good references

- http://www.python.org/doc
  - Module reference
- Google group: comp.lang.python
- Books:
  - *Python Cookbook* (recipes online)
  - *How to Think Like a Computer Scientist* (online)
- ESRI gp scripting forum

>>> Further reading

> *Python Magazine* - online print and pdf (for fee)
> *The Python Papers* - pdf free online
> *Dive Into Python* - free online

Presentation and code available at:

http://code.google.com/p/csvimporter

chad_cooper@swn.com