



Esri International User Conference | San Diego, CA
Technical Workshops | Tuesday, July 12th

Python: Integrating Standard and Third-Party Libraries

Jason Scheirer

Other Sessions

- Building Tools with Python (Technical Workshop)
 - Thursday 10:15AM 9
- Getting Started With Map Algebra Using the Raster Calculator and Python (Demo Theater Presentation)
 - Wednesday 10:00AM Exhibit Hall C (Spatial Analysis Demo Theater)
- Python - Analyzing your GPS tracking data (Demo Theater Presentation)
 - Thursday 11:00AM Exhibit Hall C (Spatial Analysis Demo Theater)
- Python - Automating Geodatabase Administration (Tech Workshop 20 Minute)
 - Thursday 11:05AM 23 B
- Python - Creating an ArcGIS Script Tool (Demo Theater Presentation)
 - Tuesday 4:00PM Exhibit Hall C (Spatial Analysis Demo Theater)
- Python - Getting Started (Technical Workshop)
 - Tuesday 1:30PM 2
 - Thursday 8:30AM 2

Other Sessions (continued)

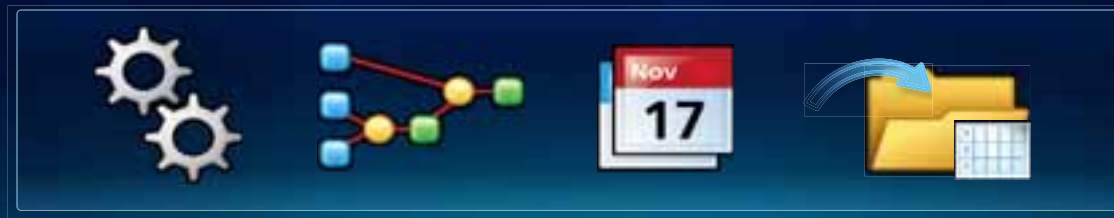
- Python - Getting to know the Python Window (Demo Theater Presentation)
 - Tuesday 10:00AM Exhibit Hall C (Spatial Analysis Demo Theater)
- Python - Improving the Performance of your Script Tools (Tech Workshop 20 Minute)
 - Tuesday 8:30AM 6 A
- Python - Raster Analysis (Technical Workshop)
 - Tuesday 3:15PM 6 C
 - Wednesday 3:15PM 5 A/B
- Python - Scripting for Map Automation (Technical Workshop)
 - Tuesday 10:15AM 9
 - Wednesday 3:15PM 9
- Python - Spatial Analysis (Intermediate) (Technical Workshop)
 - Tuesday 3:15PM 1 A/B
 - Thursday 10:15AM 2

Other Sessions (yes, there are more)

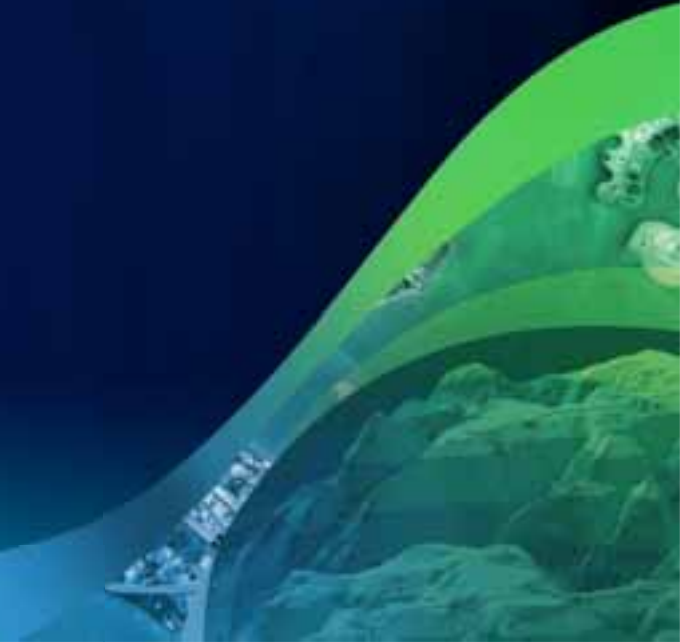
- Python - The Ease and Power of Cursors (Demo Theater Presentation)
 - Tuesday 11:30AM Exhibit Hall C (Spatial Analysis Demo Theater)
- Python Scripts - Using A Template (Demo Theater Presentation)
 - Tuesday 11:00AM Exhibit Hall C (Spatial Analysis Demo Theater)
- Road Ahead - Python Scripting Abilities
 - Thursday 4:05PM 6 B

Python in ArcGIS

- **Implementing geoprocessing script tools**
 - Creating an ArcGIS Script Tool (Tuesday 4:00PM)
 - Building Tools with Python (Thursday 10:15AM)
- **Automating workflows**
 - Gluing together other GP tools
 - Scheduled tasks
- **Integrating ArcGIS with other systems, other systems into ArcGIS**
 - Data interop: import/export
 - Automating ArcGIS processes from other applications
 - Preparing data from ArcGIS for consumption by other applications



Built-in Libraries

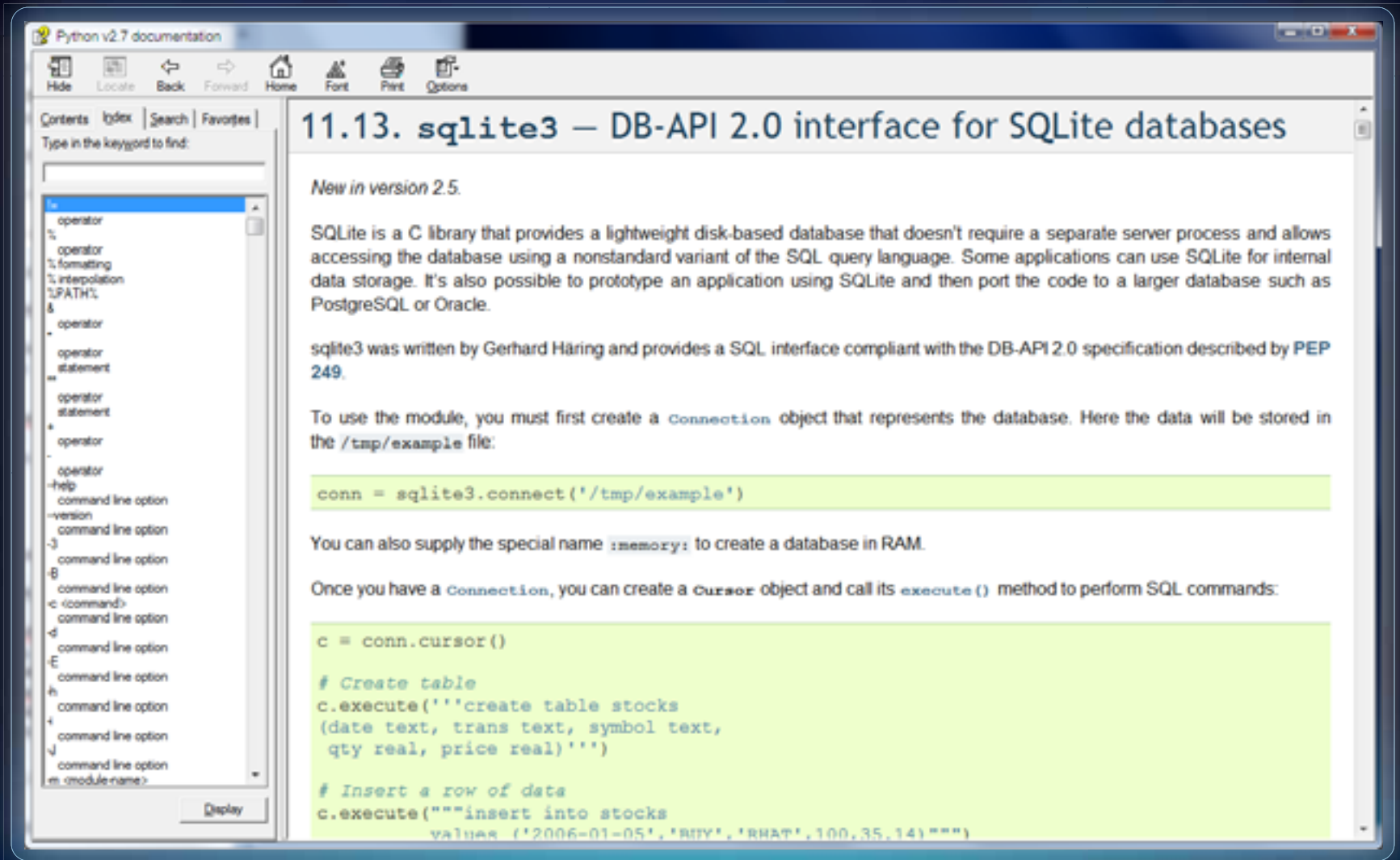


Batteries Included

- Libraries for many common tasks come with the language
 - File formats: XML (DOM, SAX, ElementTree), JSON, CSV, .ZIP (zipfile)
 - Networking: HTTP (urllib, urllib2), TCP/UDP (socket)
 - Parallelism: threading, multiprocessing, subprocess
 - Command line applications: optparse
 - Paths, directories, files: sys, os, os.path
 - User Interfaces: tkinter
 - Data structures and algorithms: collections, itertools, heapq, struct, many more
 - Date/time (time, datetime)
 - Calling C++/C (ctypes)

Read the Docs!

- Help file: ArcGIS 10.0 | Python 2.6 | Python Manuals



The screenshot shows a web browser window titled "Python v2.7 documentation". The left sidebar contains a "Contents" menu with a search bar and a list of topics. The main content area displays the "11.13. sqlite3 — DB-API 2.0 interface for SQLite databases" page. The page includes a "New in version 2.5." note, a description of SQLite, information about the sqlite3 module's origin, and usage instructions with code examples.

11.13. sqlite3 — DB-API 2.0 interface for SQLite databases

New in version 2.5.

SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

sqlite3 was written by Gerhard Häring and provides a SQL interface compliant with the DB-API 2.0 specification described by [PEP 249](#).

To use the module, you must first create a `Connection` object that represents the database. Here the data will be stored in the `/tmp/example` file:

```
conn = sqlite3.connect('/tmp/example')
```

You can also supply the special name `:memory:` to create a database in RAM.

Once you have a `Connection`, you can create a `Cursor` object and call its `execute()` method to perform SQL commands:

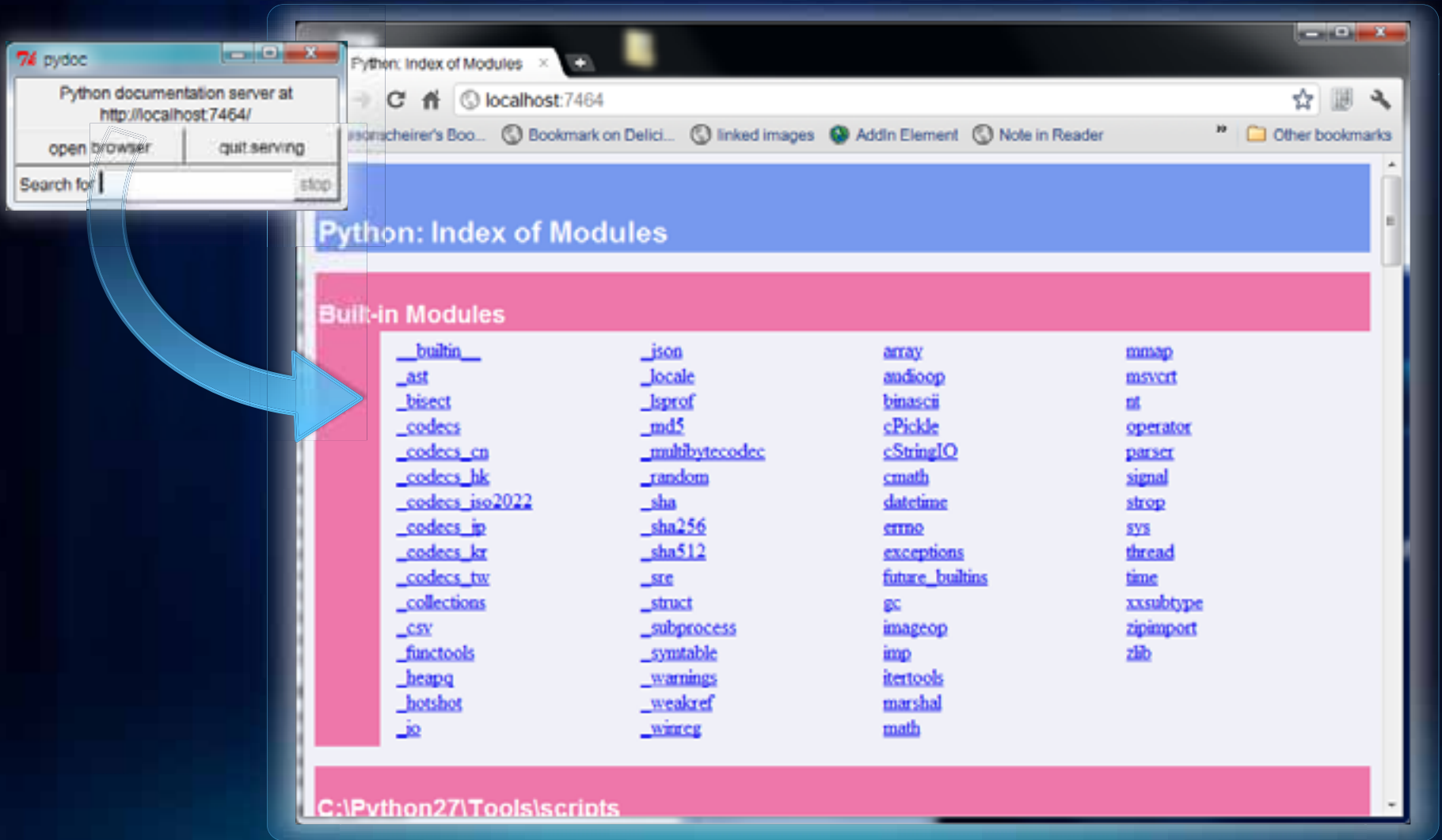
```
c = conn.cursor()

# Create table
c.execute('create table stocks
(date text, trans text, symbol text,
qty real, price real)')

# Insert a row of data
c.execute("""insert into stocks
values ('2006-01-05','BUY','RHAT',100,35.14)""")
```


Read the Docs!

- What's installed: ArcGIS 10.0 | Python 2.6 | Module Docs



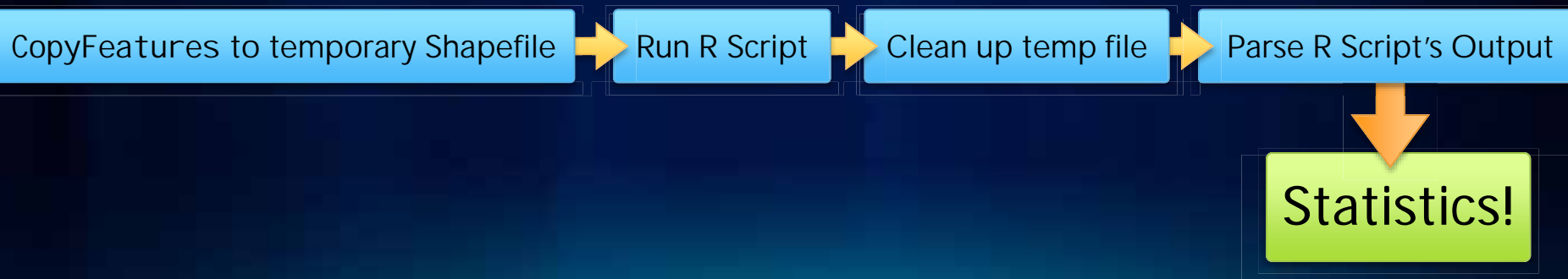
More Help

- PyMOTW (Python Module of the Week)
 - <http://www.doughellmann.com/PyMOTW/>
 - Also available as a book



Example

- **Summary Statistics in R**
 - Uses a third-party module for reading spatial data: maptools
 - Only accepts shapefiles!
- **Going to use two different system libraries:**
 - subprocess (Open/monitor other programs)
 - re (Regular Expressions)

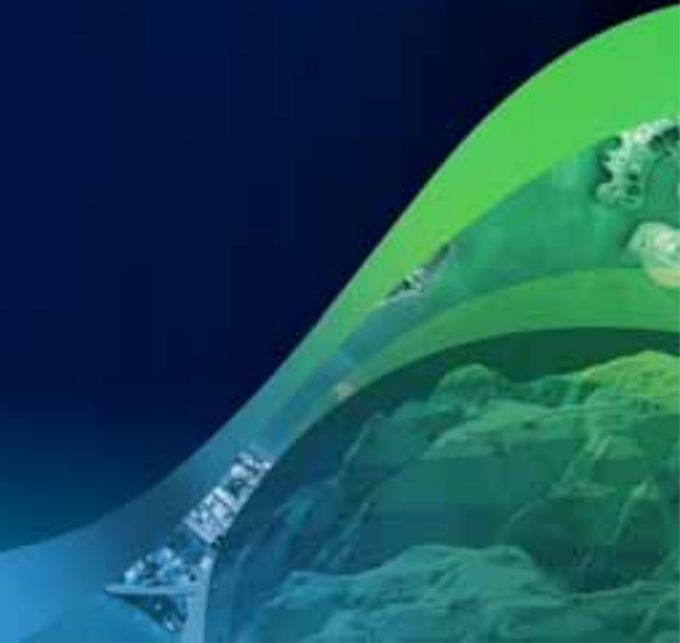


Demo

But don't reinvent the wheel!

- **wget**: Popular utility to download resources over HTTP/FTP
- Do I need it?
 - `url lib2` can download simple URLs
- But I need to follow links
 - Don't write a web spider in Python, use the `-r` option on `wget`
 - Use subprocess, like in the previous example

Third-Party Libraries



The Cheese Shop

- <http://pypi.python.org/pypi>
- The official place for 3rd party Python modules

The screenshot shows the PyPI website interface. At the top is the Python logo and a search bar. Below the logo is the text "PyPI - the Python Package Index". The main content area is divided into several sections: "PACKAGE INDEX" with links like "Browse packages", "Package submission", etc.; "Get Packages" with instructions on how to use packages; "Package Authors" with instructions on how to submit packages; and "Infrastructure" with information on interoperating with the index. A table of recent updates is also visible, listing packages like zerokspot.recipe.glt, AppDispatch, pmbot, puka, django-debian, sphinx themes bizstyle, django-easyfilters, and django-seahorse.

PyPI - the Python Package Index

The Python Package Index is a repository of software for the Python programming language. There are currently **15611** packages here. To contact the PyPI admins, please use the [Get help](#) or [Bug reports](#) links.

Get Packages

To use a package from this index either "pip install package" ([get pip](#)) or download, unpack and "python setup.py install" it. [Browse all packages](#) or use the search box above.

Package Authors

To submit a package use "python setup.py upload" ([Full tutorial](#)). The index also [hosts documentation](#). You may submit packages using [SSH](#) or the [web form](#). You must [register](#).

Infrastructure

To interoperate with the index use the [JSON](#), [XML-RPC](#) or [HTTP](#) interfaces. Use the [rsync infrastructure](#) to download even if the primary site is unavailable.

Updated	Package	Description
2011-07-06	zerokspot.recipe.glt 0.6.0	Simple zc.buildout recipe for sources in a git repository
2011-07-06	AppDispatch 0.3.4	Builds upon a PipeStack architecture to provide a nice way of writing apps.
2011-07-06	pmbot 1100b4	IRC bot - full featured, yet extensible and customizable
2011-07-06	puka 0.0.2	Puka - the opinionated RabbitMQ client
2011-07-06	django-debian 0.6	Debian integration for Django applications
2011-07-06	sphinx themes bizstyle 0.1.1	A sphinx theme for Business style Documentation.
2011-07-06	django-easyfilters 0.2	Easy creation of link-based filtering for a list of Django model objects.
2011-07-06	django-seahorse 1.0.1	Simple sandboxes that isolates production applications from

Some Interesting Modules

- PDF writing – reportlab
- Image manipulation – PIL
- (Messy) HTML – beautiful soup
- Advanced graphing/plotting – matplotlib
- Advanced date/time handling – datetime
- Excel
 - Read – xlrd
 - Write – xlwt
- Consuming .Net
 - <http://pythonnet.sourceforge.net/>
- .docx Read/write – docx
 - <https://github.com/mikemaccana/python-docx>
 - This brings us to...

There's No Installer!

- Find python.exe in your install, add it to %PATH%. Scripts\ too.
- Try pip:
 - <http://python-distribute.org/>
- Or on the command line:
 - python.exe setup.py install
- Look for it on an unofficial Python library builds site:
 - <http://www.lfd.uci.edu/~gohlke/pythonlibs/>

Conclusion

- A lot of what you need is already there
- Amazing documentation at your fingertips, more online
- There are built-in ways of interfacing with non-Python utilities (great for using script tools to integrate into ArcGIS)
- There's a HUGE ecosystem out there that probably already has what you need

