

Python Tutorial 3: Useful Libraries

Chris Waters

Python Standard Library



Operating System Interface

- *os* module
- Access to operating system functionality
 - File information/manipulation
 - Pathname operations (*os.path*)
 - Process information/management
 - Various system-specific information

System-Specific Parameters and Functions

- *sys* module
- Access to variables used by the interpreter
 - Argument list: *sys.argv*
 - Module information
- Functions that interact with the interpreter
 - Exit from Python: *sys.exit()*

Regular Expressions

- *re* module
- Compiled Expressions (*re.compile*)
- Searching (*re.search*, *re.match*, *re.findall*)
- Substitution (*re.sub*)

Math

- *math* module
- Provides access to mathematical functions defined by the C standard
 - Power and Logarithmic
 - Trigonometric
 - Angular Conversion

Internet Access

- *urllib2* module
- Functions and classes that help when opening URLs
- *urllib2.urlopen*: returns a file-like object of the given URL

Data Compression

- Common data archiving and compression formats
 - *zlib, gzip, bzip2, zipfile, tarfile*
- Different interfaces/capabilities for each format

Performance Measurement

- Timing Python code
- Small segments: *Timer*
- Code profiler: *profile*, *pstats*
- ‘High Performance’ logging profiler: *hotshot*

Testing

- *doctest* module
- Verify that functions work the way they were intended to
- Tests written into function docstrings
 - Look like interactive python sessions
- Test all functions: *doctest.testmod()*

MORE!!!one

- Web apps: *cgi*
- Audio processing
- Cryptography: *hmac, md5, sha*
- GUI: *Tkinter, wxPython, PyQt, pyKDE, pyGTK*
- Python Language Services: *parser, token*



Packages



Installing Packages

- Installer: setup.py
 - Script using *setuptools* package
- Usage: python setup.py install
 - *Sometimes* there are before/after steps

XML



XML Libraries

- Standard Library: *xml.dom*, *xml.sax*
- Packages: *ElementTree*, *cElementTree*
- <http://effbot.org/zone/element-index.htm>

Numeric



Numeric

- Fast multidimensional array functionality
- Array Objects
- Universal Functions
 - Operate on arrays
- Convenience Functions
 - Manipulate arrays

Python Imaging Library (PIL)



PIL

- Image processing library
 - Extensive file format support
- Efficient internal storage of image data
- Powerful image processing



Functionality

- Image Creation: *Image.open*
- Resizing: *img.resize*
- Rotation: *img.rotate*
- Conversion: *img.convert*
 - 1, L, P, RGB, RGBA, CMYK, YCbCr, I, F

PyOpenGL



PyOpenGL

(for C programmers)

- `void foo(int count, const int *args);`
 - `foo(args) -> None`
- `void bar(int args[4]);`
 - `bar() -> args`
 - `glGet`, etc

Array Handling

- Pointer function names
 - `glXPointer{ub | b | us | s | ui | i | f | d}`
 - `glVertexPointer(size, type, stride, pointer) -> None`
 - `glVertexPointerub(pointer[]) -> None`
 - `glVertexPointerb(pointer[]) -> None`
 - ...

Image Routines

- Similar to pointer functions
 - `glDrawPixels(width, height, format, type, pixels) -> None`
 - `glDrawPixelsub(format, pixels[...]) -> None`
 - etc...

Python Tutorial 3: Useful Libraries

Chris Waters