PTN-202: Advanced Python Programming

Course Description

This 4-day course picks up where Python I leaves off, covering some topics in more detail, and adding many new ones, with a focus on enterprise development. This is a hands-on programming class. All concepts are reinforced by informal practice during the lecture followed by lab exercises. Many labs build on earlier labs, which helps students retain the earlier material.

Audience: Advanced users, system administrators and web site administrators who want to use Python to support their server installations, as well as anyone else who wants to automate or simplify common tasks with the use of Python scripts.

Prerequisites: All students should be able to write simple Python scripts, using basic data types, program structures, and the standard Python library.

Duration: 4 days

Course Outline

1. Python Refresher
   - Variables
   - Python Data Types
   - Sequences
   - Mapping Types
   - Program Structure
   - Files and Console I/O
   - Conditionals
   - Loops
   - Defining a Function
   - Function Parameters Builtins

2. OS Services
   - The os Module
   - Environment Variables
   - Launching external programs
   - Paths, Directories and Filenames
   - Walking Directory Trees
   - Dates and Times
   - The Time Module
   - The Calendar Module
3. **Pythonic Programming**
   - Common Python Idioms
   - Slicing and Dicing
   - Unpacking Function Arguments
   - Lambda Functions
   - Nested Functions
   - List Comprehensions
   - Iterables
   - Generator Expressions
   - Writing Generators
   - Python Time Travel
   - Three Python Easter Eggs
   - A String Trick
   - String Formatting

4. **Modules and Packages**
   - Modules
   - Using `import`
   - Initialization Code
   - Namespaces
   - Executing Modules as Scripts
   - Packages
   - Configuring Import with `__init__.py`
   - Name Resolution (AKA scope)
   - Nested Functions
   - Python Style

5. **Classes**
   - About OO Programming
   - Defining Classes
   - Constructors
   - Instance Methods
   - Properties
   - Class Methods and Data
   - Static Methods
   - “Private” Methods
   - Inheritance
   - Untangling the Nomenclature
6. **Metaprogramming**
   - Special Attributes
   - `globals()` and `locals()`
   - Working with Attributes
   - The `inspect` module
   - Decorators
   - Decorator Functions
   - Decorator Classes
   - Decorating Classes
   - Creating Classes at Runtime
   - Monkey Patching

7. **Objectives**
   - Program Development
   - Comments
   - **Pylint**
     - Customizing `pylint`
   - Using `pyreverse`
     - The `unittest` module
   - Skipping Tests
   - Making a suite of tests
   - Automated test discovery
   - Using **Nose**
     - The Python Debugger
     - Starting debug mode
     - Stepping through a Program
     - Setting Breakpoints
   - Profiling
   - Benchmarking

8. **Distributing Modules**
   - Installing Packages
   - Ways to distribute code
   - Overview of `distutils`
   - Preparing for distribution
   - Creating a source distribution
   - Creating built distributions
   - **Setup.py** Options
   - **Setup.py** Commands
   - Code Portability
9. **Database Access**
   - The DB API
   - Available Interfaces
   - Connecting to a Server
   - Creating a Cursor
   - Executing a Statement
   - Fetching Data
   - Tip: Making an iterator for large queries
   - Parameterized Statements
   - Dictionary Cursors
   - Metadata
   - Transactions
   - Object-relational Mappers

10. **qt GUI Programming with PyQt4**
    - What is PyQt4?
    - Event Driven Applications
    - GUI Application Flow Chart
    - External Anatomy of a PyQt4 Application
    - Internal Anatomy of a PyQt4 Application
    - Using designer
    - Anatomy of a designer-based application
    - Naming Conventions
    - Common Widgets
    - Layouts
    - Selectable Buttons
    - Making your application stretch
    - Actions and Events
    - Menu Bar
    - Status Bar
    - Using predefined dialogs
    - Creating Custom Dialogs
    - Tabs
    - Niceties
    - Working with Images

10.tk **GUI Programming with Tkinter**
    - Objectives
    - Tkinter Overview
• Basic Tkinter Programming
• Object-oriented Tkinter
• Widgets
• Labels
• Buttons
• Setting Fonts
• Colors
• Standard Colors
• Variable Wrappers
• Selectable Buttons
• Text Entry Blanks
• Multiline Text Entry/Display
• Listbox
• Arranging Widgets
• Using `pack()`
• Tweaking the layout
• Frames
• Adding Scrollbars
• Callbacks
• Callback Parameters
• Binding Events
• Event Specifications
• Creating Menus

11. Network Programming
• Sockets
• Socket options
• Client Concepts
• Server Concepts
• Application Protocols
• Forking Servers
• Grabbing HTML from the Web
• Consuming Web Services
• Web Data the Easier Way
• Sending email
• Binary Data
• The `struct` module

12. Multiprogramming
• What are Threads?
• The Python Thread Manager
• The `threading` module
• Threads for the impatient
• Creating a thread class
• Variable Sharing
• Using Queues
• Debugging threaded programs
• The multiprocessing module
• Alternatives to multiprogramming

13. System Administration
• The `subprocess` module
• `subprocess` convenience functions
• Using the `sh` module
• Remote access
• Other useful modules
• Permissions
• Saving Information
• Creating a useful command line script
• Creating Filters
• Parsing the command line
• Simple logging
• Logging Levels
• Formatting Log Entries
• Logging to other Destinations

14. XML and JSON
• About XML
• Normal approaches to XML
• Which module to use?
• Getting Started with `ElementTree`
• How `ElementTree` works
• Creating a new XML Document
• Parsing an XML Document
• Navigating the XML Document
• Using XPath
• About JSON
• Reading JSON
• Writing JSON

15. Extending Python
• Why extend Python?
• Ways to extend Python with C
• Hand-coded C
• Overview
• The C Program
• Methods
• The Method Table
• The init function
• Handling errors
• Custom exception objects
• Putting it all together
• Using SWIG
• The interface file
• Generating the Wrappers
• Building and installing the extension

• cytpes

Appendix A: Books

Appendix B: String Formatting