Course Summary

Description

This advanced course covers both advanced development techniques such as metaprogramming, and cutting-edge technologies such as git and hadoop. Students completing this course will be able to create sophisticated modules and frameworks that can be leveraged by other programmers. As with all our programming classes, this custom Python course has a generous complement of exercises that allow the student to immediately practice new concepts. Students will write numerous real-world Python scripts, which manipulate real-world data.

Because of the advanced audience, the instructor may cover other topics as requested by students.

Topics

- Unit testing (unittest, pytest, nose, pymock)
- git/github (gitlib)
- Remote processes (paramiko, pexpect)
- Advanced XML (validation, schema generation, namespaces)
- Emulating container classes (dict, list)
- Parsing complex text (pyparsing)
- Design patterns
- Web scraping (scrapy, BeautifulSoup)
- Functional tools (single-dispatch, lru_cache, map/reduce)
- Advanced metaprogramming (descriptors, __new__)
- Virtual Python environments (virtualenv, pyenv)
- Cloud computing (Hadoop, etc)
- Packaging and distributing modules

Audience

This course is appropriate for experienced Python programmers who want to learn very advanced Python skills. The course content can be applied to a wide range of tasks.

Prerequisites

Before taking this course, students should be comfortable writing intermediate Python scripts, using basic and complex data types, program structures and the standard Python library. In addition, they should be able to create basic classes, and understand intermediate concepts such as decorators and generators. This is not a class for beginners.

Duration

Five days
ProTech Professional Technical Services, Inc.

Extreme Python v3

Course Outline

I. Virtual environments
   A. Why virtual?
   B. Using virtualenv
   C. Addons for virtualenv
   D. Using pyenv
   E. Pitfalls of virtual environment

II. Design Patterns
   A. What are design patterns?
   B. Abstract factory
   C. builder
   D. facade
   E. flyweight
   F. command
   G. mediator
   H. Others as suggested by students

III. Packaging and distributing modules
   A. The Python packaging morass
   B. Tarballs, eggs, and wheels, oh my!
   C. Preparing a package for distribution
   D. Types of distributions
   E. Installing from distributions
   F. Creating standalone executables

IV. Functional tools
   A. About functional programming
   B. The functools module
   C. single-dispatch
   D. caching
   E. map/reduce
   F. partial functions

V. Container Classes
   A. Built-in containers
   B. Other containers in the standard library
   C. Creating list-like containers
   D. Creating dict-like containers
   E. Free-form containers

VI. Advanced iterables
   A. What is an iterable?
   B. Generator expressions and functions
   C. Iterators
   D. The iterator protocol
   E. Making classes iterable

VII. Advanced Metaprogramming
    A. callable classes
    B. descriptors
    C. __new__
    D. metaclasses

VIII. Advanced XML
     A. lxml recap
     B. Schema validation
     C. Working with namespaces
     D. Generating data structures from schemas
     E. Generating schemas from data structures

IX. Mako templates
   A. What is a template
   B. Mako templates
   C. Mako syntax
   D. Power features
   E. Creating templates for executable jobs

X. Web scraping
   A. Web scraping concepts
   B. Using requests
   C. Parsing *ML with BeautifulSoup
   D. Automating scraping with Scrapy

XI. PyParsing for markup
    A. About pyParsing
    B. Defining a tag grammar
    C. Attaching code
    D. Parsing *ML

XII. Unit Testing
     A. Types of unit test
     B. unittest module
     C. Using pytest
     D. doc tests
     E. Features of nose

XIII. Remote access
      A. Executing remote programs
      B. Remote file transfer
      C. Scripting remote sessions

XIV. Git/GitHub
     A. About gitlib
     B. Creating a local repository
     C. Adding and updating files
     D. Working with github

XV. Workflow
    A. Creating processing pipelines
    B. Using celery
    C. Using luigi

XVI. Cloud computing
     A. About the cloud
     B. Specific cloud technologies
     C. Hadoop