

Specialized - Applied Python for Scientists & Engineers

Code:	TTPS4870
Length:	5 days
URL:	View Online

Throughout the course students will be led through a series of progressively advanced topics, where each topic consists of lecture, group discussion, comprehensive hands-on lab exercises, and lab review. This course is “skills-centric”, designed to train attendees in essential Python and web development skills, coupling the most current, effective techniques with the soundest coding practices.

Skills Gained

- Create and run basic programs
- Design and code modules and classes
- Implement and run unit tests
- Use benchmarks and profiling to speed up programs
- Process XML and JSON
- Manipulate arrays with numpy
- Get a grasp of the diversity of subpackages that make up scipy
- Use iPython notebooks for ad hoc calculations, plots, and what-if?
- Manipulate images with PIL
- Solve equations with sympy

Who Can Benefit

While there are no programming prerequisite, programming experience is helpful. Students should be comfortable working with files and folders, and should not be afraid of the command line.

Course Details

The Python Environment

- About Python
- Starting Python
- Using the interpreter
- Running a Python script
- Python scripts on Unix/Windows
- Using the Spyder editor

Getting Started

- Using variables
- Builtin functions
- Strings
- Numbers
- Converting among types
- Writing to the screen
- String formatting
- Command line parameters

Flow Control

- About flow control
- White space
- Conditional expressions (if,else)
- Relational and Boolean operators
- While loops
- Alternate loop exits

Sequences

- About sequences
- Lists and tuples
- Indexing and slicing
- Iterating through a sequence
- Sequence functions,keywords, and operators
- List comprehensions
- Generator expressions
- Nested sequences

Working with files

- File overview
- Opening a text file
- Reading a text file
- Writing to a text file
- Raw (binary) data

Dictionaries and Sets

- Creating dictionaries
- Iterating through a dictionary
- Creating sets
- Working with sets

Functions

- Defining functions
- Parameters
- Variable scope
- Returning values
- Lambda functions
- Exporting images

Errors and Exception Handling

- Syntax errors
- Exceptions
- Using try/catch/else/finally
- Handling multiple exceptions
- Ignoring exceptions

OS Services

- The os module
- Environment variables
- Launching external commands
- Walking directory trees
- Paths, directories, and filenames
- Working with file systems
- Dates and times

Pythonic idioms

- Small Pythonisms
- Lambda functions
- Packing and unpacking sequences
- List Comprehensions
- Generator Expressions

Modules and packages

- Initialization code
- Namespaces
- Executing modules as scripts
- Documentation
- Packages and name resolution
- Naming conventions
- Using imports

Classes

- Defining classes

- Constructors
- Instance methods and data
- Attributes
- Inheritance
- Multiple inheritance

Developer tools

- Analyzing programs with pylint
- Creating and running unit tests
- Debugging applications
- Benchmarking code
- Profiling applications

XML and JSON

- Using ElementTree
- Creating a new XML document
- Parsing XML
- Finding by tags and XPath
- Parsing JSON into Python
- Parsing Python into JSON

iPython

- iiPython basics
- Terminal and GUI shells
- Creating and using notebooks
- Saving and loading notebooks
- Ad hoc data visualization

numpy

- numpy basics
- Creating arrays
- Indexing and slicing
- Large number sets
- Transforming data
- Advanced tricks

scipy

- What can scipy do?
- Most useful functions
- Curve fitting
- Modeling
- Data visualization

- Statistics

A tour of scipy subpackages

- Clustering
- Physical and mathematical Constants
- FFTs
- Integral and differential solvers
- Interpolation and smoothing
- Input and Output
- Linear Algebra
- Image Processing
- Distance Regression
- Root-finding
- Signal Processing
- Sparse Matrices
- Spatial data and algorithms
- Statistical distributions and functions
- C/C++ Integration

pandas

- pandas overview
- Dataframes
- Reading and writing data
- Data alignment and reshaping
- Fancy indexing and slicing
- Merging and joining data sets

matplotlib

- Creating a basic plot
- Commonly used plots
- Ad hoc data visualization
- Advanced usage

The Python Imaging Library (PIL)

- PIL overview
 - Core image library
 - Image processing
 - Displaying images
-

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy

Get Your Free Copy Now

ExitCertified® Corporation and iMVP® are registered trademarks of ExitCertified ULC and ExitCertified Corporation and Tech Data Corporation, respectively
Copyright ©2021 Tech Data Corporation and ExitCertified ULC & ExitCertified Corporation.
All Rights Reserved.

Generated 9