QUICKSTART

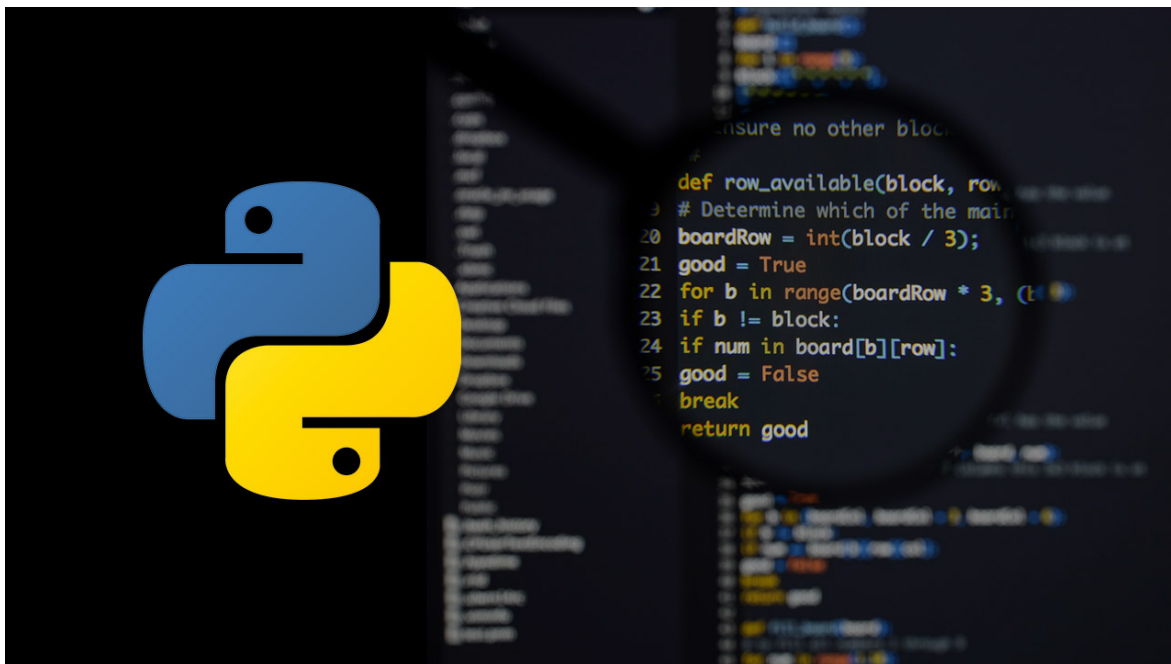Over 35 Years Of Technology Training

Document Generated: 12/18/2025

Learning Style: Virtual Classroom

Technology:

Difficulty: Intermediate

Course Duration: 5 Days

Applied Python for Scientists and Engineers (TTPS4870)

About this Course:

This intermediate-level training program is specifically designed for Python Programmers, Data Scientists, and Data Engineers. This course provides a comprehensive overview of Applied Python for professionals with fundamental knowledge of Python Programming. This course provides such a professional an incredible learning opportunity to master the core concepts of Python Programming and use its feature for seamless mathematical & scientific computing.

This course overviews Fundamental Python Concepts and Nurture Scripting Skills

required to Create & Run Python Programs. Professionals will also get to learn Module & Classes Designing, Unit Test Implementation, Profiling & Benchmarking, JSON & XML Processing, SciPy Classes, NumPy Array Manipulation, PIL Image Manipulation, and many more key concepts. This course will help professionals develop a conceptual understanding of key modules of Python Programming for Scientists & Engineers. On average, a professional Python Data Scientist earns $93,185 annually.

Course Objective:

The core objective of this course is to help professionals gain a better knowledge and sound understanding of the following key concepts:

- Basic Programs Creation & Running
- Modules & Classes Designing and Programming
- Unit Test Implementation & Performance
- Speeding Up Programs with Profiling and Benchmarks
- JSON and XML Processing and NumPy Array Manipulation
- SciPy Classes of Packages & Sub Packages
- Ad-Hoc Plots, What-Ifs, & Calculations with iPython Notebook
- PIL Image Manipulation and SymPy Equation Solution
- PySpark and SciKit-Learn Advanced Data Science Concepts

Audience:

This course is specifically tailored for the following group of professionals and interested candidates:

- Python Programmers
- Data Scientists & Data Engineers
- IT Professionals & Experts

Prerequisites:

There are no obligatory prerequisites for the Applied Python for Data Science (TTPS4870) Course. However, it is highly recommended for professionals to have fundamental knowledge & experience of programming.

Course Outline:

Module 1: The Python environment

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

Module 2: Getting started

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

Module 3: Flow control

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

Module 4: Sequences

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

Module 5: Working with files

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

Module 6: Dictionaries and sets

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

Module 7: Functions

- Defining functions
- Parameters

- Variable scope
- Returning values
- Lambda functions

Module 8: Errors and exception handling

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

Module 9: OS services

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

Module 10: Pythonic idioms

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

Module 11: Modules and packages

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

Module 12: Classes

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

Module 13: 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

Module 14: iPython and Jupyter

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

Module 15: numpy

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

Module 16: scipy

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

Module 17: 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

Module 18: pandas

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

Module 19: Seaborn

- Creating a basic plot
- Commonly used plots
- Ad hoc data visualization
- Exporting images

Module 20: matplotlib

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

Module 21: The Python Imaging Library (PIL) [Optional]

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

Module 22: SciKit-Learn Essentials [Optional]

- SciKit overview
- SciKit-Learn overview
- Algorithms Overview
- Classification, Regression, Clustering, and Dimensionality Reduction
- SciKit Demo

Module 23: PySpark Overview [Optional]

- Python and Spark
- SciKit-Learn vs. Spark MLlib
- Python at Scale
- PySpark Demo