Document Generated: 11/14/2026

Learning Style: Online Instructor-Led

Technology:

Difficulty: Beginner

Course Duration: 3 Days

Python Fundamentals for Data Science (TTPS4874)



About This Course:

Geared for scientists and engineers with limited practical programming background or experience, Python Fundamentals for Data Science is a hands-on introductory-level course that provides you with a ramp-up to using Python for scientific and mathematical computing. Working in a hands-on learning environment with Jupyter notebooks, you'll learn basic Python scripting skills and concepts, as well as the most important Python modules for working with data, from arrays, to statistics, to

plotting results.

Throughout the course, guided by our expert instructor, you'll gain a robust skill set that will equip you to make data-driven decisions and elevate operational efficiencies within your organization. You'll explore data manipulation with Pandas, advanced data visualization using Matplotlib, and numerical analysis with NumPy. You'll also delve into best practices for error and exception handling, modular programming techniques, and automated workflow development, equipping you with the skill set to enhance both the effectiveness and efficiency of your data-driven projects.

Course Objectives:

- Core Python Proficiency: By the close of the course, participants will have a firm grasp on the foundational elements of Python, such as variables, data types, and flow control, empowering them to write scripts and build simple programs with confidence.
- Analytical Problem-Solving: Utilizing libraries such as NumPy and SciPy, students will develop the ability to perform complex mathematical operations and statistical analyses, significantly amplifying their analytical capabilities for tasks such as data modeling or optimization problems.
- Data Manipulation Mastery: By the end of the course, participants will be proficient in employing Pandas to clean, transform, and analyze data sets, enabling them to make data-driven decisions effectively.
- Automated Workflow Development: Students will acquire the ability to construct automated scripts using Python's Standard Library, optimizing repetitive tasks and thereby enhancing operational efficiency in their organizations.
- Advanced Data Visualization: Upon course completion, learners will be equipped to utilize Matplotlib and other Python libraries to craft intricate visual representations of data, facilitating clearer and more impactful reporting and presentations.
- Error-Resilient Coding: Attendees will learn best practices for implementing robust error and exception handling techniques, leading to the creation of more stable and secure Python applications.
- Modular Programming Proficiency: By mastering Python functions, modules, and packages, students will be adept at developing modular and maintainable code, a key skill for scalability and collaborative programming projects.

Audience:

- This introductory-level course is geared for technical professionals new to

Python. Roles include data analysts, developers, engineers or anyone tasked with utilizing Python for data analytics tasks. Familiarity with basic scripting skills is recommended, as this course does not teach general scripting basics.

Prerequisites:

- Familiarity with basic scripting skills is recommended, as this course does not teach general scripting basics.

Course Outline:

Getting Started with the Python Environment

- Starting Python
- Using the interpreter
- Running a Python script
- Editors and IDEs

iPython and Jupyterlab

- iPython features & iPython "magic" commands
- iPython configuration
- Creating Jupyter notebooks
- Managing notebooks with Jupyterlab

Variables and Values

- Using variables
- Builtin functions
- String data
- Numeric data
- Converting types

Basic input and output

- Writing to the screen
- String formatting
- Command line arguments
- Reading the keyboard

Flow Control

- About flow control
- The if statement
- Relational and Boolean values
- while loops
- Exiting from loops

Array types

- Sequence types in general
- Lists and list methods
- Tuples
- Indexing and slicing
- Iterating through a sequence
- Sequence functions, keywords, and operators
- List comprehensions and generators

Working with files

- File I/O overview
- Opening a text file
- Reading a text file
- Writing to a text file

Dictionaries and Sets

- About dictionaries
- Creating dictionaries
- Getting values
- Iterating through a dictionary
- About sets
- Creating sets
- Working with sets

Functions, modules, and packages

- Returning values
- Types of function parameters
- Variable scoping
- Documentation best practices
- Creating and importing modules
- Organizing modules into packages

Intro to Pandas

- Pandas overview
- Series and Dataframes
- Reading and writing data
- Data summaries
- Data alignment and reshaping
- Selecting and indexing

- Basic Data Plotting

Pandas Part 2

- Merging and joining data sets
- Categorical data
- Time series and dates
- Working with strings
- Pandas options

Matplotlib

- Creating a basic plot
- Commonly used plots
- Ad hoc data visualization
- Leveraging Seaborn for better plots
- Exporting images

Additional Topics (Time Permitting)

The following chapters are included for extended coverage and may be addressed as time allows. Their inclusion depends on class pacing, participant engagement, and time availability.

Intro to NumPy (Time permitting)

- NumPy basics
- Creating arrays
- Indexing and slicing
- Large number sets
- Transforming data
- SciPy overview

Introduction to AI with Python for Data Analysis (Time permitting)

- Overview of AI Libraries
- Setting Up Your Environment:
- Understanding AI Models
- Creating Your First Model
- Evaluating Model Performance

Practical AI Projects in Python (Time permitting)

- Set up a Python project for AI applications.
- Data Handling
- Model Development
- Test and validate your AI model's effectiveness.
- Applying Your Model

Excel spreadsheets (Time permitting)

- The openpyxl module
- Reading an existing spreadsheet
- Creating a spreadsheet from scratch
- Modifying an existing spreadsheet

Serializing Data (Time permitting)

- Parsing JSON into Python
- Parsing Python into JSON
- Working with CSV
- Creating a new XML document
- Parsing XML
- Searching XML by tags and XPath

Jupyter Widgets (Time permitting)

- What are widgets?
- Implementation via iPython
- Creating callbacks
- Implementing handlers