

Course duration

- 5 days

Course Benefits

- Create and run basic Python programs.
- Design and code modules and classes.
- Implement and run unit tests.
- Use benchmarks and profiling to speed up programs.
- Work with various data formats.
- Process XML and JSON.
- Manipulate arrays with NumPy.
- Get a grasp of the diversity of subpackages that make up SciPy.
- Leverage pandas to easily create and structure data.
- Use matplotlib to create amazing visualizations.
- Use Jupyter notebooks for ad hoc calculations, plots, and what-if?.

Course Outline

1. The Python Environment
 1. Starting Python
 2. Using the Interpreter
 3. Running a Python Script
 4. Python Scripts on Unix
 5. Python Scripts on Windows
 6. Python Editors and IDEs
2. Getting Started
 1. Using Variables
 2. Built-in Functions
 3. Strings
 4. Numbers
 5. Converting among Types
 6. Writing to the Screen
 7. String Formatting
 8. Command Line Parameters
3. Flow Control
 1. About Flow Control
 2. What's with the White Space?
 3. if and else
 4. Conditional Expressions
 5. Relational Operators

- 6. Boolean Operators
- 7. while Loops
- 8. Alternate Ways to Exit a Loop
- 4. Lists and Tuples
 - 1. About Sequences
 - 2. Lists
 - 3. Tuples
 - 4. Indexing and Slicing
 - 5. Iterating through a Sequence
 - 6. Functions for All Sequences
 - 7. Operators and Keywords for Sequences
 - 8. Nested Sequences
 - 9. List Comprehensions
 - 10. Generator Expressions
- 5. Working with Files
 - 1. Text file I/O
 - 2. Opening a Text File
 - 3. Reading a Text File
 - 4. Writing to a Text File
 - 5. "Binary" (Raw, or Non-delimited) Data
- 6. Dictionaries and Sets
 - 1. About Dictionaries
 - 2. When to Use Dictionaries
 - 3. Creating Dictionaries
 - 4. Iterating through a Dictionary
 - 5. About Sets
 - 6. Creating Sets
 - 7. Working with Sets
- 7. Functions
 - 1. Defining a Function
 - 2. Function Parameters
 - 3. Variable Scope
 - 4. Returning Values
 - 5. Lambda Functions
- 8. Exception Handling
 - 1. Syntax Errors
 - 2. Exceptions
 - 3. Handling Exceptions with Try
 - 4. Handling Multiple Exceptions
 - 5. Handling Generic Exceptions
 - 6. Ignoring Exceptions
 - 7. Using else
 - 8. Cleaning Up with finally
 - 9. Re-raising Exceptions
 - 10. Raising a New Exception
- 9. OS Services
 - 1. The os Module

- 2. Environment Variables
- 3. Launching External Commands
- 4. Paths, Directories, and Filenames
- 5. Walking Directory Trees
- 6. Dates and Times
- 10. Modules and Packages
 - 1. Initialization code
 - 2. Namespaces
 - 3. Executing modules as scripts
 - 4. Documentation
 - 5. Packages and name resolution
 - 6. Naming conventions
 - 7. Using imports
- 11. Classes
 - 1. Defining Classes
 - 2. Constructors
 - 3. Instance methods and data
 - 4. Attributes
 - 5. Inheritance
 - 6. Multiple Inheritance
- 12. Programmer Tools
 - 1. Program Development
 - 2. Comments
 - 3. pylint
 - 4. Customizing pylint
 - 5. Unit Testing
 - 6. The unittest Module
 - 7. Creating a Test Class
 - 8. Establishing Success or Failure
 - 9. Startup and Cleanup
 - 10. Running the Tests
 - 11. Debugging
 - 12. Benchmarking
 - 13. Profiling Applications
- 13. Excel Spreadsheets
 - 1. openpyxl module
 - 2. Reading an Existing Spreadsheet
 - 3. Creating a Spreadsheet
 - 4. Modifying a Spreadsheet
- 14. XML and JSON
 - 1. Creating XML Files
 - 2. Parsing XML
 - 3. Tags and XPath
 - 4. Reading and Writing JSON
- 15. iPython and Jupyter
 - 1. About iPython and Jupyter
 - 2. iPython Basics

- 3. Jupyter Basics
- 16. NumPy
 - 1. Python's scientific Stack
 - 2. NumPy Overview
 - 3. Creating Arrays
 - 4. Creating Ranges
 - 5. Working with Arrays
 - 6. Shapes
 - 7. Slicing and Indexing
 - 8. Indexing with booleans
 - 9. Stacking
 - 10. Iterating
 - 11. Tricks with Arrays
 - 12. Matrices
 - 13. Data Types
 - 14. NumPy Functions
- 17. SciPy
 - 1. About SciPy
 - 2. SciPy Packages
 - 3. SciPy Examples
- 18. pandas
 - 1. About pandas
 - 2. Series
 - 3. DataFrames
 - 4. Reading and Writing Data
 - 5. Indexing and Slicing
 - 6. Merging and Joining Data Sets
- 19. matplotlib
 - 1. Creating a plot
 - 2. Commonly Used Plots
 - 3. Customizing Styles
 - 4. Ad hoc data visualization
 - 5. Advanced Usage
 - 6. Saving Images

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *would be useful* for this Python class:

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