1-800-633-1440
www.mindshare.com
training@mindshare.com

Python Programming

Training

Let MindShare Bring "Python Programming" to Life for You

The *Python Programming* course examines the programming techniques required to develop Python software applications as well as integrate Python to a multitude of other software systems. The core course covers statistical data analysis in Python using NumPy/SciPy, Pandas, Matplotlib (publication-quality scientific graphics) that mirror support functions in R.

You Will Learn:

- Python development and optimization
- Python Function protocols
- Dynamic typing and memory model
- Python object-oriented features
- Built-in data types, sequence types (lists, dictionaries, tuples, etc), and type operators (slicing, comprehensions, generators, etc)
- Building and using libraries and packages

Who Should Attend?

The target audience includes programmers, system administrators and QA engineers, with little or no knowledge of Python programming. You will be able to develop, automate, and test applications and systems using one of the most powerful programming languages.

Course Length: 4 Days (but customizable to shorter duration)

Course Outline:

The curriculum explores topics in current and emerging Python modules and in-depth usage examples.

Schedule:

- Day 1: Getting started, basics, types, control structures, data structures
- Day 2: Intermediate to advanced Python: object orientation, regular expressions
- Day 3: NumPy/SciPy, Pandas, Matplotlib, and data integration
- Day 4: Python Advanced: Interoperability of Microsoft Excel, Python Database API

Day 1: Introduction to Python

IPython (ipython) Notebook Overview
  - Getting started with notebook infrastructure: starting, editing, running, and note taking
  - Executing a Python program
  - Python interpretation and object system
- Python Program Development
  - Syntax: indentation, code blocks, colon separation
  - Comments, strings, numbers, arithmetic operators, arithmetic expressions
  - Functions, modules, import, reload, function arguments (default, keyword), nesting
  - Boolean expressions, relational and logical operators
  - Control structures: if, elif, else, for loop, while, range operator
  - Dynamic typing and namespaces
- Structured Types: Sequences
  - Strings, operations, methods, slicing, and formatting
  - List operations, methods, slicing, iteration, sorting, and comprehension
  - Dictionaries (hash table): indexing, creating, iterating, and sets

1-800-633-1440
www.mindshare.com
training@mindshare.com

  - Tuples and sets and their management
  - Extended datatypes: defaultdict, ordereddict, namedtuple
- System Utilities
  - File I/O support (reading, writing) and file scanners, standard input/output
  - OS system support (module os)
  - Command line parsing of arguments (module: optparse)
  - Exception handling & catching exceptions
  - Embedding program start: _name_and_'_main_'
  - Subprocess and multiprocessing modules (module: subprocess)
- Packages
  - Python packages
  - Practical file/directory search and manipulation (modules: shutil, tempfile, cwd)

Day 2: Intermediate to Advanced Python

- Regular Expressions
  - Basic regular expression syntax patterns and matching (module: re)
  - Extraction and slicing
  - Substitutions
  - Named groups
- Object-orientated Programming
  - Classes, class fields, iterators, and attributes
  - Properties, methods, functions as objects (indirect calls), and operator overloading
  - Inheritance, customization, operator overloading, class variables
  - Persistent objects with shelf and pickling (modules: pickle, shelve)
  - Iteration support:
- Advanced Function Design
  - Unpacking returned sequences
  - Functional techniques: lambda, map/filter/reduce, list.sort,
  - Generators
  - Decorators
- Built-in Tools and Testing
  - Debugging options
  - Timing and profiling Python programs
  - Optimization patterns for higher performance
  - Test generation (module: unittest)

Day 3: Numerical processing with Python: NumPy/SciPy, Pandas

- NumPy/SciPy Overview
  - Open-source modules to Python for common mathematical and numerical routines
  - Demonstration of common pre-compiled and fast functions (use cases)
  - Special functions in SciPy
- NumPy Arrays
  - Array mathematics, iteration, basic array operations
  - Array item selection and manipulation
  - Vector and matrix mathematics
  - Statistics and random numbers
- Matplotlib- visualization
  - Generating plots and figures from Python data
  - Plot variations and attributes
- Pandas and data integration
  - Data frames
  - Aggregation and grouping of data
  - Reshaping, transforming, and cleaning of data
  - Scraping data: web API, parsing html and XML, JSON (encoder/decoder)
  - Time series analysis

1-800-633-1440
www.mindshare.com
training@mindshare.com

Day 4: Advanced Python Development - Python Integration

- Python Interoperability with Microsoft Excel Workbooks and Data
  - Read (xlrd), write (xlwt)
  - Utilities (xlutils)
- Python Database Integration
  - Using “dbm” databases
  - Object persistence: shelves
  - Pickling objects without shelves
- Python Database API to External Databases
  - Acquiring Python database connection and setup
  - SQL fundamentals
  - Creating a table, adding data, reading data
  - Integration example: using Python with MySQL Database
  - Integration example: MongoDB and PyMongo
  - Bottle: Python Web Framework
- Python Machine Learning Module [Optional Machine Learning]
  - Clustering: Hierarchical Clustering, Memory-saving Hierarchical Clustering, k-means
  - Regression: Least Squares, Support Vector Machines (SVM), Least Squares (LS)
  - Classification: k-Nearest-Neighbor, Linear Discriminant Analysis (LDA), Basic Perceptron
- Performance Optimization and Parallel Programming
  - Array-oriented computing
  - Examples: image Processing, vectorization, and time-series analysis
  - Array-based computing with Cylon - Optimizing static compiler
  - Numba- Dynamic compiler: supporting the creation of Python extension types and GPU/Multicore support for technical computing
  - NumbaPro development of NumPy ufuncs, gufuncs, and decorators
- GUI Toolkit (Tkinter and wxPython) [Optional Interactive Python]
  - Python GUI support
  - Adding buttons, frames, and callbacks
  - Getting input from a user
  - Layout attributes
  - Building and reusing GUIs by sub-classing

Customization: The course design has options to customize subject material to meet specific goals with concepts such as:

- Numba is a just-in-time specializing compiler that compiles annotated Python and NumPy code to LLVM (through decorators) and can be used to target high-performance multicore and GPU implementations (such as NVIDIA CUDA/OpenCL)
- Python database integration and web techniques including: SQL, MongoDB, Bottle
- Python Machine Learning module (mlpy)
- GUI Toolkit (Tkinter and wxPython) [Optional Interactive Python]

Python Programming Hands-on Experience

- IPython Notebook environment– online environment for coding and recording Python programs and note taking on concepts. Participants save code development in electronic on-line notebooks that are universally accessible from other python systems. As intended, individuals are not required to have numerous software packages locally on their laptops or workstations. Instead cloud-based software enables all students to explore Python programming irrespective of their individual computing preference (Windows, Mac, Linux). The course examples are presented in IPython Notebook that allows participants to explore all of the instructor’s lecture examples.
- Hands-on exercises– self-contained exercise modules using real-world data are presented to give the necessary and the practical experience to participating students.

1-800-633-1440
www.mindshare.com
training@mindshare.com

Recommended Prerequisites:

Prior programming experience and familiarity with programming concepts such as variables/scopes, flow-control, and functions. Object-oriented programming concepts are not required, but definitely beneficial.

Course Material:

Mindshare will supply electronic version of the 300+ page lecture set that includes lecture materials, example source code, coding exercises with solutions, supplemental examples, and tools. Coverage of over 30 most common and advanced Python modules including os, sys, shutil, tempfile, subprocess, glob, profile, shelve, optparse, unittest.