

# Introduction to Python

## Course 408 – 40 Hours

### Overview

Python is an easy to learn and powerful programming language that is used in many computer science areas.

This five-day workshop concentrates on the Python programming language and many libraries. From basic procedural syntax to sophisticated object-oriented programming techniques, delegates will learn how to write Python scripts and applications with code that is robust, maintainable and efficient.

The course is presented as a mixture of lectures and hands-on exercises. Practical sessions follow each topic, designed to reinforce the points covered. Additional information is provided in appendices to extend the learning experience after the course has been completed.

### Course Objectives

#### **On Completion, students will be able to**

- Execute Python code in a variety of environments
- Use correct Python syntax in Python programs
- Use the correct Python control flow construct
- Write Python programs using various collection data types
- Write home grown Python functions
- Use many of the standard Python modules such as os, sys, math, and time
- Trap various errors via the Python Exception Handling model
- Use the IO model in Python to read and write disk files
- Create their own classes and use existing Python classes
- Understand and use the Object Oriented paradigm in Python programs
- Use the Python Regular Expression capabilities for data verification
- Use many scientific libraries provided by python

### Who Should Attend

This course is designed for anyone who needs to learn how to write programs in Python

### Course Contents

- **Introduction to Python**
  - Strengths and Weaknesses
  - A Brief History of Python
  - Python Versions
  - Installing Python
  - Environment Variables

- Executing Python from the Command Line
- IDLE
- Editing Python Files
- Getting Help
- Dynamic Types
- Python Reserved Words
- Naming Conventions
  
- **Basic Python Syntax**
  - Introduction
  - Basic Syntax
  - Comments
  - String Values
  - String Operations
  - The format Method
  - String Slices
  - String Operators
  - Numeric Data Types
  - Conversions
  - Simple Input and Output
  - The print Function
  
- **Language Components**
  - Introduction
  - Control Flow and Syntax
  - Indenting
  - The if Statement
  - Relational Operators
  - Logical Operators
  - True or False
  - Bit Wise Operators
  - The while Loop
  - break and continue
  - The for Loop
  
- **Collections**
  - Introduction
  - Lists
  - Tuples
  - Sets
  - Dictionaries
  - Sorting Dictionaries
  - Copying Collections
  - Summary
  
- **Functions**
  - Introduction
  - Defining Your Own Functions

- Parameters
- Function Documentation
- Keyword and Optional Parameters
- Passing Collections to a Function
- Variable Number of Arguments
- Scope
- Functions - "First Class Citizens"
- Passing Functions to a Function
- Mapping Functions in a Dictionary
- Lambda
- Closures
  
- **Modules**
  - Overview
  - Standard Modules - sys
  - Standard Modules - math
  - Standard Modules - time
  - The dir Function
  
- **Exceptions**
  - Errors
  - Run Time Errors
  - The Exception Model
  - Exception Hierarchy
  - Handling Multiple Exceptions
  - raise
  - assert
  - Writing Your Own Exception Classes
  
- **Input and Output**
  - Introduction
  - Data Streams
  - Creating Your Own Data Streams
  - Access Modes
  - Writing Data to a File
  - Reading Data From a File
  - Additional File Methods
  - Using Pipes as Data Streams
  - Handling IO Exceptions
  - Working with Directories
  - Metadata
  - The pickle Module
  
- **Classes in Python**
  - Principles of Object Orientation
  - Creating Classes
  - Instance Methods
  - File Organization

- Special Methods
- Class Variables
- Inheritance
- Polymorphism
- Type Identification
- Custom Exception Classes
- Class Documentation - pydoc
  
- **Regular Expressions**
  - Introduction
  - Simple Character Matches
  - Special Characters
  - Character Classes
  - Quantifiers
  - The Dot Character
  - Greedy Matches
  - Grouping
  - Matching at Beginning or End
  - Match Objects
  - Substituting
  - Splitting a String
  - Compiling Regular Expressions
  - Flags
  
- **Introduction to NumPy and SciPy**
  - Python arrays and NumPy Arrays
  - Multi dimensional Arrays
  - Array slicing
  - Fancy Indexing
  - Data types
  - Array calculation methods
  - Statistics methods
  - Universal functions
  - Broadcasting
  - Universal function methods
  - Matplotlib
  - Line plots
  - Scalar plots
  - Bar plots
  - Histograms
  - Multiple plots
  - Image display
  - SciPy overview
  - Interpolation
  - Integration
  - FFT
  - Signal and image processing
  - Optimizations

- Statistics
- Linear algebra
- Matrix objects
- Introduction to IPython
- Interactive work
- Numerical computing with IPython
- Interactive Plotting
- High performance and parallel computing
- Debugging with pydev