

This three day course emphasizes becoming productive quickly in the Python programming language. This course starts by covering the basic Python language syntax including data types, variables, operators, conditionals and flow control. From there, students will learn how to create more modular applications using features such as functions, modules, and classes. The class ends with coverage of regular expressions, file input/output, and exceptions.

**Course Objectives:**

- Create and invoke Python programs.
- Select the most appropriate data types and data structures for Python variables.
- Improve program design and modularity with functions and modules.
- Implement object-oriented techniques with Python classes and objects.
- Use regular expression pattern matching to search and manipulate complex data.
- Read and write data files using Python’s input and output operators.
- Write programs which both handle and raise exceptions.

**Audience:** Programmers and system administrators who are new to the Python language.

**Prerequisites:** Previous experience programming in at least one other programming language is beneficial.

**Number of Days:** 3 days

- |  |   |
|--|---|
| <p><b>1. Meet Python</b><br/> Origin of Python<br/> Where to Get Python<br/> Invoking Python<br/> Other Python Implementation<br/> Python Resources</p>                      | <p><b>5. Conditional Expressions</b><br/> Conditions and Conditional Expressions<br/> Numeric Comparisons and Conditions<br/> Comparing Strings<br/> Complex Conditions</p> |
| <p><b>2. Handling Numbers and Strings</b><br/> Numeric Data<br/> Using Numbers<br/> String Data<br/> Using Strings</p>   | <p><b>6. Basic Input and Output</b><br/> Reading Input<br/> Simple Output<br/> Formatted Output</p>   |
| <p><b>3. Variables and Data Structures</b><br/> Variables<br/> Numeric and String Variables<br/> Lists<br/> Tuples<br/> Sets<br/> Dictionaries<br/> Additional Resources</p> | <p><b>7. Function</b><br/> Creating Functions<br/> Invoking Functions<br/> Returning Values from Functions<br/> Passing Parameters</p>                                      |
| <p><b>4. Flow Control</b><br/> Controlling the Flow</p>  |   |

More About Function Arguments

Scope of Variables

Lambda Functions

**8. Using Modules**

What are Modules?

Creating Modules

Using Modules

More About Modules

Compiled Modules

Standard Modules

**9. An Introduction to Classes**

Namespaces

What is a Class?

Defining a Class

Defining Class and Instance Attributes

Defining Methods

Instantiating a Class

Inheritance

Resources

**10. Regular Expressions**

Pattern Matching vs. Wildcards

Compiling Regular Expressions

Matching vs. Searching

Substitution

Regular Expressions: Metacharacters

The Problem with \

Special Backslash Sequences

Using MatchObject

Additional Regular Expression

Operation

Additional Resources

**11. Files**

File Handles and File-like Objects

Opening and Closing Files

Reading from Files

Writing to Files

Moving Around in Files

The pickle Module

Resources

**12. Errors and Exceptions**

Errors, Exceptions, and Warnings

Raising an Exception

User Defined Exceptions

Catching and Handling Exceptions

The raise Statement