

Introduction to Python

Length: 3 Days

Pre-requisites: Students should have some experience with at least one programming language. Typically, students in this course will have already programmed in C, C++, Java, Perl, Ruby, VB, or anything equivalent to these languages.

Summary: This course introduces the student to the Python language. Upon completion of this class, the student will be able to write non trivial Python programs dealing with a wide variety of subject matter domains. Topics include language components, the IDLE environment, control flow constructs, strings, I/O, collections, classes, modules, and regular expressions. The course is supplemented with many hands on labs using either Linux or Windows.

Upon completion of this course, 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

Target Audience: This course is designed for anyone who needs to learn how to write programs in Python.

COURSE CONTENT

An Introduction to Python

- 1.Introductory Remarks about Python
- 2.Strengths and Weaknesses
- 3.A Brief History of Python
- 4.Python Versions
- 5.Installing Python
- 6.Environment Variables

- 7.Executing Python from the Command Line
- 8.IDLE
- 9.Editing Python Files
- 10.Getting Help
- 11.Dynamic Types
- 12.Python Reserved Words
- 13.Naming Conventions

Basic Python Syntax

- 1.Introduction
- 2.Basic Syntax
- 3.Comments
- 4.String Values
- 5.String Operations
- 6.The format Method
- 7.String Slices
- 8.String Operators
- 9.Numeric Data Types
- 10.Conversions
- 11.Simple Input and Output
- 12.The print Function

Language Components

- 1.Introduction
- 2.Control Flow and Syntax
- 3.Indenting
- 4.The if Statement
- 5.Relational Operators
- 6.Logical Operators
- 7.True or False
- 8.Bit Wise Operators
- 9.The while Loop
- 10.break and continue
- 11.The for Loop

Collections

- 1.Introduction
- 2.Lists
- 3.Tuples
- 4.Sets
- 5.Dictionaries
- 6.Sorting Dictionaries

7.Copying Collections

8.Summary

Functions

- 1.Introduction
- 2.Defining Your Own Functions
- 3.Parameters
- 4.Function Documentation
- 5.Keyword and Optional Parameters
- 6.Passing Collections to a Function
- 7.Variable Number of Arguments
- 8.Scope
- 9.Functions - "First Class Citizens"
- 10.Passing Functions to a Function
- 11.Mapping Functions in a Dictionary
- 12.Lambda
- 13.Closures

Modules

- 1.Modules
- 2.Standard Modules - sys
- 3.Standard Modules - math
- 4.Standard Modules - time
- 5.The dir Function

Exceptions

- 1.Errors
- 2.Run Time Errors
- 3.The Exception Model
- 4.Exception Hierarchy
- 5.Handling Multiple Exceptions
- 6.raise
- 7.assert
- 8.Writing Your Own Exception Classes

Input and Output

- 1.Introduction
- 2.Data Streams
- 3.Creating Your Own Data Streams
- 4.Access Modes
- 5.Writing Data to a File
- 6.Reading Data From a File
- 7.Additional File Methods
- 8.Using Pipes as Data Streams
- 9.Handling IO Exceptions
- 10.Working with Directories
- 11.Metadata
- 12.The pickle Module

Classes in Python

- 1.Classes in Python
- 2.Principles of Object Orientation
- 3.Creating Classes
- 4.Instance Methods
- 5.File Organization
- 6.Special Methods
- 7.Class Variables
- 8.Inheritance
- 9.Polymorphism
- 10.Type Identification
- 11.Custom Exception Classes
- 12.Class Documentation - pydoc

Regular Expressions

- 1.Introduction
- 2.Simple Character Matches
- 3.Special Characters
- 4.Character Classes
- 5.Quantifiers
- 6.The Dot Character
- 7.Greedy Matches
- 8.Grouping
- 9.Matching at Beginning or End
- 10.Match Objects
- 11.Substituting
- 12.Splitting a String
- 13.Compiling Regular Expressions
- 14.Flags