



Python Basics

Lecture and Lab

5 Day Course

Python is an interpreted, object-oriented, high-level language that can get work done in a hurry. A tool that can improve all professionals ability to do work, Python is freely available on early all major platforms without a charge.

Python Basics

- **Recommended Follow up: *Advanced Python (5 days)***
- **Recommended Follow up: *Python for Network Automation (5 days)***

1. Introduction to Python

- Installing Python 2.x
- Installing Python 3.x
- Preparing to write Python
- Preparing to write a Python file (*.py) - Text Editors
- Executing a Python file
- Python Enhancement Proposals
- How to “speak” Python
- Python Statements and Control Flow
- Python Interpreter

2. Version Controlling Code

- Set up a github account
- Github essentials
- README course requirements
- How to Set up a repo
- Issue a Pull request
- How to use “Issues” for peer review

3. Basics of Programming

- Built in Functions
- Functions and Arguments

- Print()
- White spacing basic rules
- String Literal Escape Sequences
- Python Variables
- Naming Conventions & Rules
- Types as Objects
- Variable References & Garbage Collection
- Sequence Types
- Membership Statements
- List Iteration
- List Enumeration
- Sequence Assignments
- Mutable vs Immutable Objects
- Multi Target Assignments
- Arithmetic Expressions
- Assignment Shorthand

4. Python Basic Variables and Data Types

- Numeric Types
- Operators and Precedence
- Integers
- Floating points

5. String Types

- Generating Strings in Python
- Common String Methods
- Formatting String Output
- Booleans
- Printing and formatting strings
- Scripting with input()
- About raw_input()

6. Python Lists & Tuples

- Intro to Integers
- Mixed variable lists
- Tuples
- Common List Methods
- Other List Operations
- Tuples

- Python Time Tuples
- Tuples vs Lists
- Python Dictionaries
- Python Dictionaries
- Keys and Values
- Dictionary Building
- Assigning Values to Dictionaries
- Dictionary Methods
- Dictionaries vs Lists & Tuples

7. If, elif, else

- Relational Operators
- Logical operators
- Comparison Operations
- “simple” if Statement
- If else statements
- If elif
- Nested if statement

8. Looping with “while”

- While usage
- Count controlled loop
- Event controlled loop
- Continue
- Break

9. Looping with “for”

- The for Loop
- For iteration example

10. Understanding Iterators

- The range() Function
- Taking the range() of len()
- Iterable Objects
- The iter() Function
- Looping with dictionaries
- Looping with lists

11. Basic Input/Output with Files

- Opening Files
- Working with Files
- Read data from files
- Controlling Output Location

12. Python List Comprehension

- Basic List Comprehensions
- Compound List Comprehensions

13. Creating Python Functions

- Function Basics
- Defining Functions
- Function Polymorphism
- Argument Defaults
- Lambdas
- Local Variables
- Variable Masking
- Creating functions using Sorted() and sort()
- Preventing Variable Modifications
- Argument Matching Methods

14. Modules & Packages

- Pip and pip3
- Module Basics
- Packages
- Virtual environments
- Defined modules
- Import modules
- From import statements
- Namespace

15. Python Scope

- Naming conventions
- Local scope
- Global scope
- Nested scope

16. Object Oriented Python

- About OOP

- The Class Statements
- Defining a class
- Class Inheritance
- Classes as Objects
- Using Dictionaries
- Understanding self
- Class fields and constructors
- Data structures
- Sub classes (Inheritance)
- Multiple Inheritance
- Static methods