Python 3 Programming Training

BILGINC IT Academy
info@bilginc.com
bilginc.com

Duration: **4 Day**

- **Build and run real Python 3 applications from the ground up.**
- **Designed for programmers adopting Python 3 or migrating existing Python 2 applications.**
- **Hands-on labs are delivered in practical coding environments throughout the course.**

Python 3 Programming is a technical course designed for developers who want to learn the Python 3 programming language through a combination of instructor-led teaching and practical coding exercises.

The hands-on activities are delivered using MS Windows, although the Python concepts and programming techniques covered throughout the course are equally applicable to other platforms, including Linux.

Participants begin with the fundamentals of Python and progress through data types, flow control, string processing, collections, regular expressions, file handling, functions, modules, packages, and Object-Oriented Programming. Later sections cover exception handling, processes and threads, multiprocessing, and the Python Standard Library.

The course also helps developers moving from Python 2 to Python 3 understand important differences between the two versions.

Prerequisites

No previous Python experience is required. However, participants should already be familiar with at least one programming language.

Experience with another scripting language, such as Perl or PHP, is beneficial. Previous exposure to Object-Oriented Programming will also be useful when progressing into the more advanced sections of the course.

Who Should Attend

This course is suitable for:

- Programmers beginning new development projects with Python 3
- Developers migrating existing Python 2 applications to Python 3

Python 3 Programming Training

BILGINC IT Academy
info@bilginc.com
bilginc.com

Duration: **4 Day**

What You Will Learn

By the end of the course, participants should be able to:

- Use the Python interactive interpreter to write and execute Python 3 programs.
- Understand the key language elements of Python 3.
- Make effective use of Python's extensive collection of libraries and modules.
- Recognize important differences between Python 2 and Python 3.
- Identify simple and complex variable types and select suitable types for different requirements.
- Work with Python 3 operators and built-in functions.
- Apply procedural control flow in Python 3.
- Perform file input and output, including working with persistent data objects.
- Organize code effectively using functions and modules.
- Apply Object-Oriented Programming techniques in Python 3.
- Build more robust applications using error trapping and reporting.
- Run and control external programs from Python.
- Use multiprocessing to take advantage of multi-core processors.

Training Outline

Chapter 1 – Introduction to Python 3

- What is Python?
- What is Python 3?
- Why use Python?
- Performance downsides
- The Python community
- Running Python interactively
- Python scripts
- Python help
- Anatomy of a Python script
- Modules
- Functions and built-ins

Chapter 2 – Fundamental Variables

- Python as an Object-Oriented language
- Python variables
- Variable names
- Type-specific methods
- Operators and types
- Augmented assignments
- Python types
- Switching between types
- Introduction to Python lists
- Introduction to Python tuples
- Introduction to Python dictionaries

Chapter 3 – Flow Control

BILGINC IT Academy
info@bilginc.com
bilginc.com

- Python conditionals
- Indentation
- Understanding truth
- Boolean and logical operators
- Chained comparisons
- Sequence and collection tests
- Object types
- Introduction to Exception Handling
- While loops
- Loop control statements
- For loops
- enumerate
- Counting for loops
- Zipping multiple lists
- Conditional expressions
- Unconditional flow control

Chapter 4 – String Handling

- Python 3 strings
- The print function
- Creating and processing strings
- String concatenation
- Quotes
- String methods
- String tests
- String formatting
- Additional formatting techniques
- String slicing
- split and join methods

Chapter 5 – Collections

- Review of Python types
- Useful tuple operations
- Python lists
- Tuple and list slicing
- Extended iterable unpacking
- Adding items to lists
- Removing items by position
- Removing list items by content
- Sorting
- List methods
- Sets
- Making effective use of sets
- Set operators
- Python dictionaries
- Dictionary values
- Removing dictionary items
- Dictionary methods
- View objects

BILGINC IT Academy
info@bilginc.com
bilginc.com

Chapter 6 – Regular Expressions

- Python regular expressions
- Extended RE meta-characters
- Regular expression objects
- Regular expression substitution
- Regular expression splitting
- Matching alternatives
- Anchors
- Class shortcuts
- Flags
- Repeat quantifiers
- Quantifiers
- Parentheses groups
- Back-references
- Global matches

Chapter 7 – Data Storage and File Handling

- File objects
- Reading files into Python
- File reading techniques
- Filter programs with the fileinput module
- Binary mode
- Writing files from Python
- Standard streams
- Additional file handling techniques
- Random access
- Python pickle persistence
- Pickle protocols
- Building shelves
- Compression
- Database interface overview
- Example: SQLite with Python

Chapter 8 – Functions

- Python functions
- Function parameters
- Variadic functions
- Default parameter values
- Named / keyword parameters
- Enforcing named parameters
- Returning objects from functions
- Variables within functions
- Nested functions
- Variables in nested functions
- Function documentation
- Lambda functions
- Lambda as a sorting key
- Lambda with re.sub

BILGINC IT Academy

info@bilginc.com

bilginc.com

Chapter 9 – Advanced Collections

- Advanced list operations with filter
- List comprehensions
- Set and dictionary comprehensions
- Lazy lists
- Generators
- Generator objects and next
- Coroutines
- Using list comprehensions as generators
- Collection copying challenges
- Slice-based copying
- Copying with deepcopy

Chapter 10 – Modules and Packages

- Understanding modules and packages
- Multiple source files
- How Python locates modules
- Importing modules
- Importing names
- Using directories as packages
- Writing modules
- Module documentation
- Testing modules
- Python debugger
- Python profiler
- Distributing libraries with distutils

Chapter 11 – Introduction to Classes and OOP

- Classes and OOP
- Object-Oriented terminology
- Object-Oriented Programming
- Working with objects
- Defining classes
- Defining methods
- Constructing objects
- Special methods
- Operator overload special methods
- Properties
- Properties and decorators
- Class methods
- Inheritance
- Inheritance terminology

Chapter 12 – Error Handling and Exceptions

- Writing to stderr
- Controlling warnings
- Exception handling
- Exception syntax

BILGINC IT Academy
info@bilginc.com
bilginc.com

- Multiple exceptions
- Exception arguments
- The finally block
- Order of execution
- The Python exception hierarchy
- Common mistakes
- The raise statement
- Creating and raising custom exceptions
- assert

Chapter 13 – Multitasking

- Introduction to processes
- Creating processes from Python
- Legacy interface examples
- Waiting for child processes
- The subprocess module
- The subprocess.Popen class
- Running a basic process
- Capturing output
- Passing data through a pipe
- Processes and threads
- Basic threading in Python
- Synchronization objects in threading
- Challenges associated with threads
- The multiprocessing module
- Queue objects

Chapter 14 – The Python Standard Library

- The Python Standard Library
- Example: converting Python 2 scripts to Python 3
- Pretty Printer
- Operating System interfaces: os and related tools
- System-specific attributes: sys
- Signal handling with signal
- Converting a signal into an exception
- Configuration files
- The configparser module
- datetime and related modules
- The platform module
- External function interface: ctypes
- The socket module
- __future__
- Other Standard Library modules