Maslak Mahallesi, Maslak Meydan Sk. No:5. Spring Giz Plaza. Maslak/İstanbul
+90 212 282 7700 - info@bilginc.com

# Python 3 Programming

Learn via: **Classroom / Virtual Classroom / Online**

Duration: **4 Days**

## Overview

This is a technical course that introduces the Python 3 programming language. It is an instructor led presentation and hands on exercises course using MS Windows, but is equally applicable to other platforms such as Linux.

## Prerequisites

No previous knowledge of Python is assumed, although delegates should be familiar with at least one programming language.

Experience of another scripting language, such as Perl or PHP, will be an advantage, as will previous experience of object oriented programming.

## Who Should Attend

This {*training} is suitable for programmers starting new projects in Python 3, or for those porting existing applications from Python 2.

## What You Will Learn

At the end of this {*training} you will be able to:

- Use the Python interactive interpreter to write and run Python 3 programs
- Understand Python 3 language elements
- Exploit the rich library of Python libraries and modules
- Appreciate the differences between Python 2 and Python 3
- Recognize simple and complex variable types and select appropriately
- Use Python 3 operators and built-in functions
- Understand procedural control flow in Python 3
- Program file input/output, including persistent data objects.
- Create well organized and efficient code using functions and modules
- Use Object Oriented programming techniques in Python 3.
- Build robust applications with error trapping and reporting
- Run and control other programs from Python
- Take advantage of multi-core processors with multiprocessing

## Outline

### Chapter 1: Introduction to Python 3

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

**Chapter 2: Fundamental Variables**

- Python is Object Oriented
- Python variables
- Variable names
- Type specific methods
- Operators and type
- Augmented assignments
- Python types
- Switching types
- Python lists introduced
- Python tuples introduced
- Python dictionaries introduced

**Chapter 3: Flow Control**

- Python conditionals
- Indentation
- What is truth?
- Boolean and logical operators
- Chained comparisons
- Sequence and collection tests
- Object types
- A note on Exception Handling
- While loops
- Loop control statements
- For loops
- enumerate
- Counting 'for' loops
- Zipping through multiple lists
- Conditional expressions
- Unconditional flow control

**Chapter 4: String Handling**

- Python 3 strings
- The print function
- Cooking strings
- String concatenation
- 'Quotes'
- String methods
- String tests
- String formatting
- Other string formatting aids
- Slicing a string
- String methods - split and join

**Chapter 5: Collections**

- Python types - reminder
- Useful tuple operations
- Python lists
- Tuple and list slicing
- Extended iterable unpacking
- Adding items to a list
- Removing items by position
- Removing list items by content
- Sorting
- List methods
- Sets
- Exploiting sets
- Set operators
- Python dictionaries
- Dictionary values
- Removing items from a dictionary
- Dictionary methods
- View objects

**Chapter 6: Regular Expressions**

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

**Chapter 7: Data Storage and File Handling**

- New file objects
- Reading files into Python
- Reading tricks
- Filter programs - fileinput module
- Binary mode
- Writing to files from Python
- Standard streams
- More tricks
- Random access
- Python pickle persistence
- Pickle protocols
- Build some shelves
- Compression
- Database interface overview
- Example - SQLite from Python

**Chapter 8: Functions**

- Python functions
- Function parameters
- Variadic functions
- Assigning default values to parameters
- Named (keyword) parameters
- Enforcing named parameters
- Returning objects from a function
- Variables in functions
- Nested functions
- Variables in nested functions
- Function documentation
- Lamdba functions
- Lambda as a sort key
- Lambda in re.sub

**Chapter 9: Advanced Collections**

- Advanced list functions - filter
- List comprehensions
- Set and dictionary comprehensions
- Lazy lists
- Generators
- Generator objects and next - coroutines
- List comprehensions as generators
- Copying collections - problem
- Copying collections - slice solution?
- Copying collections - deepcopy solution

**Chapter 10: Modules and Packages**

- What are modules and packages?
- Multiple source files
- How does Python find a module?
- Importing a module
- Importing names
- Directories as packages
- Writing a module
- Module documentation
- Testing a module
- Python debugger
- Python profiler
- Distributing libraries - distutils

**Chapter 11: Introduction to Classes and OOP**

- Classes and OOP
- Object-Oriented terminology
- Object-Oriented Programming
- Using objects
- Defining classes
- Defining methods
- Constructing an object
- 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
- Multiple exceptions
- Exception arguments
- The finally block
- Order of execution
- The Python exception hierarchy
- A common mistake
- The raise statement
- Raising our own Exceptions
- assert

**Chapter 13: Multitasking**

- Family life
- Creating a process from Python
- Old interface examples
- Waiting for a child
- Using the subprocess module
- The subprocess.Popen class
- Running a basic process
- Capturing the output
- Passing data through a pipe
- Processes and threads
- Very basic threads in Python
- Synchronization objects in threading
- The trouble with threads
- Using the multiprocessing module
- Queue objects

**Chapter 14: The Python Standard Library**

- The Standard Library
- Example - converting Python 2 scripts to Py3
- Pretty Printer - a useful utility
- Operating System interfaces - os and friends
- System specific attributes - sys
- Signal handling - signal
- Converting a signal to an exception
- Configuration files
- The configparser module
- The datetime module and friends
- The platform module
- External function interface - ctypes
- The socket module
- __future__
- Other modules