



# **ActiveNET®**

**#202, Manjeera Plaza, Ameerpet, HYD 9848111288**

**[www.activenetinformatics.com](http://www.activenetinformatics.com)**

**Email:[activesurya@gmail.com](mailto:activesurya@gmail.com)**

**By Suryanarayana  
20 Years Exp**

## **About Python:**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community after 30 years.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

## **Some programming-language features of Python are:**

1. A variety of basic data types are available: numbers (floating point, complex, and unlimited-length long integers), strings (both ASCII and Unicode), lists, and dictionaries.
2. Python supports object-oriented programming with classes and multiple inheritance.
3. Code can be grouped into modules and packages.
4. The language supports raising and catching exceptions, resulting in cleaner error handling.

5. Data types are strongly and dynamically typed. Mixing incompatible types (e.g. attempting to add a string and a number) causes an exception to be raised, so errors are caught sooner.
6. Python contains advanced programming features such as generators and list comprehensions.
7. Python's automatic memory management frees you from having to manually allocate and free memory in your code.



## 7 Reasons why you must consider writing Software Applications in Python

### 1) Readable and Maintainable Code

### 2) Multiple Programming Paradigms

It supports Object Oriented and Structured programming fully. Also its language features support various concepts in functional and aspect-oriented programming. At the same time Python also features a dynamic type system and automatic memory management.

### 3) Compatible with Major Platforms and Systems

### 4) Robust Standard Library

### 5) Many Open Source Frameworks and Tools

For instance, you can simplify and speedup web application development by using robust Python **Web Frameworks** like *Django, Cubic Web, Pyramid, Giotto, Pylon, Turbo Gears, Web2Py, Flask, Bottle, and CherryPy*. **Asynch Frameworks** like *Sanic, Tornado*, **Analytical Frameworks** like *Dash*, Likewise, you can accelerate **desktop GUI application development** using **Python GUI frameworks** and toolkits like *PyQT, PyJs, PyGUI, Kivy, PyGTK and WxPython*.

### 6) Simplify Complex Software Development

Python is used for both desktop and web applications and also used in complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. With this you can create custom big data solutions without putting extra time and effort. Many python developers even use it to accomplish Artificial Intelligence (AI) and Natural Language Processing (NLP).

## **7) Adopt Test Driven Development**

**UnitTest, Nose Testing, Pytest, Robot Framework, Zope Testing**

# Core Python

## **1. Python 3 – Overview**

**First Program, History of Python, Python Features**

## **2. Python 3 - Environment Setup**

**Installation of Python, Setting up PATH, Running Python, Python IDEs (Wing IDE, PyCharm, Atom, Stani's Python Editor, IDLE, Spyder, Komodo Edit, PyScripter, Geany, Thonny, KDevelop, Eclipse +PyDev, Sublime Text, Visual Studio Code, GNU Emacs, Brackets)**

## **3. Python 3 - Basic Syntax**

**First Python Program on Command Prompt, Script Mode Programming, Python Identifiers, Reserved Words, Lines and Indentations, Multi-line Statements, Quotations in Python, Comments in Python, Using Blank lines, Waiting for the User Input, Multiple Statements on a Single Line, Multiple Statement Groups as Suites, Command Line Arguments**

## **4. Python 3 - Variable Types**

**Assigning values to variables, Multiple Assignments, Standard Data types, Python Numbers, Strings, Lists, Tuple, Dictionary, Data Type conversion,**

## **5. Python 3 - Basic Operators**

**Arithmetic Operators, Comparison Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators,**

## **6. Python 3 - Decision Making**

**if statement, if-else statement, if-elif-else statement, nested if statement**

## **7. Python 3 – Loops**

While Loop, While-Else Loop, For Loop, For-Else Loop, For Loop with Range, break statement, continue statement

#### 8. Python 3 – Numbers

int, float, complex numbers, Number type conversion, Mathematical functions, Random Number Functions, Trigonometric Functions, Mathematical constants

#### 9. Python 3 – Strings

Accessing values to Strings, Updating strings, Escape characters, String Special operators, String Formatting operators, Triple Quotes, Unicode Strings

#### 10. Python 3 – Lists

Python Lists, Accessing values in List, Updating Lists, Deleting List Elements, Basic List Operations, Indexing, Slicing and Matrixes, Built-in List functions & Methods

#### 11. Python 3 – Tuples

Accessing values in Tuples, Updating Tuples, Deleting Tuple Elements, Basic Tuple operations, Indexing, slicing and Matrixes, No Enclosing Delimiters, Built-in Tuple Functions

#### 12. Python 3 – Dictionary

Accessing values in Dictionary, Updating Dictionary, Deleting Dictionary Elements, Properties of Dictionary keys, Built-in Dictionary Functions and Methods

#### 13. Python 3 - Date & Time

What s Tick? What s TimeTuple?, Getting current time, Getting formatted time, Getting Calendar for a Month, The time Module, The calendar module, Other module functions

#### 14. Python 3 – Functions

Defining a function, Calling function, Pass by Reference Vs value, Function Arguments, Required Arguments, Keyword Arguments, Default Arguments, Variable-Length Arguments, The Anonymous functions, Lambda functions, The return statement, Scope of variables, Global Vs Local variables

#### 15. Python 3 – Modules

The import statement, The from...import statement, The from...import\* statement, Executing module as scripts, Locating modules, The PYTHONPATH variable, Namespaces and Scoping, The dir() function, The globals() and locals() functions, The reload() function, Packages in Python

#### 16. Python 3 - Files I/O

Printing to the screen, Reading Keyboard Input, The input function, Opening and closing Files, The open function, The file object attributes, The close() method, Reading and writing files, The write() method, The read() method, File positions, Renaming and Deleting Files, The rename() method, The remove() method, Directories() in Python, The mkdir() method, The chdir() method, The getcwd() method, The rmdir() method, File and Directories related methods

#### 17. Python 3 - Exceptions Handling

Standard Exceptions, Assertions in Python, What is Exception?, Handling an Exception, The except clause with No Exceptions, The try-finally clause, Argument of an Exception, Raising an Exception, User-defined Exceptions

# Advanced Python

#### 18. Python 3 - Object Oriented

Overview of OOPs Terminology, Creating Classes, Creating Instance objects, Accessing Attributes, Built-in class Attributes, Destroying objects, Class Inheritance, Overriding Methods, Base overloading Methods, Overloading Operators, Data Hiding

#### 19. Python 3 - Multithreaded Programming

What is multithreading?, Starting a new Thread, The Threading Module, Creating Threads using Threading module, Synchronizing Threads, Multi-threaded Priority Queues,

#### 20. Python 3 - Network Programming

Network Programming Introduction, What is Sockets, The Socket Module, Server Socket Methods, Client Socket Methods, General Socket Methods, A Simple Server, A Simple Client, Python Internet Modules,

#### 21. Python 3 - Regular Expressions

Basic patterns that match single characters, Compilation flags, The match() function, The search() function, Matching Vs Searching, Search and Replace, Regular Expression Modifiers: Option Flags, Regular Expression Patterns, Regular Expression Examples, Special Character Classes, Repetition Cases, Non-greedy repetition, Grouping with Parenthesis, Back references, Alternatives, Anchors, Special syntax with Parentheses

## 22. Python 3 - MySQL Database Access

What is PyMySQL?, How do I install PyMySQL?, Database connection, Creating Database Table, INSERT operation, READ operation, UPDATE operation, DELETE operation, Performing Transactions, COMMIT operation, ROLLBACK operation, Disconnecting Database, Handling Errors,

## 23. Python 3 - CGI Programming

What is CGI?, Web Browsing, CGI Architecture Diagram, Web Server Support and Configuration, First CGI Program, HTTP Header, CGI Environment Variables, GET and POST methods, Passing information using GET method, Simple FORM GET method example, Passing information using POST method, Passing checkbox data to CGI program, Passing Radio button data to CGI program, Passing TextArea data to CGI Program, Passing Drop Down Box data to CGI Program, Using Cookies in CGI, Setting p Cookies, Retrieving Cookies, File Upload Example, How to raise a "File Download" Dialog Box

## 24. Python 3 - XML Processing

What is XML?, XML Parser Architectures and APIs, Parsing XML with SAX API, The make\_parser() method, The parse() method, The parseString() method, Parsing XML with DOM API

## 25. Python 3 - Sending Email using SMTP

SMTP Introduction, Sending an HTML email using Python, Sending attachments as an Email,

## 26. Python 3 - GUI Programming (Tkinter)

Tkinter, wxPython, PyQt, JPython, Tkinter Programming, Tkinter Widgets, Standard Attributes, Geometry management

## 27. Introduction to Python Frameworks

**Best of Luck**