

Introduction to Python

Python is a widely-used open source general-purpose scripting language. Python is an object-oriented, high-level programming language with dynamic semantics. Python is simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules which encourages program modularities and code reuse. Debugging Python programs is easy a bug or bad input will never cause a segmentation fault.

Why Python

Python runs on Windows, Linux/Unix, Mac OS and has been ported to Java and .NET virtual machines. Python is free to use, even for the commercial products, because of its OSI-approved open source license. Python is an example of an open source software. In simple terms, users can freely distribute copies of the software, read the source codes, make the changes to it and use pieces of it in new free programmes. Open source is based on the concept of a community which shares knowledge. This is one of the best reasons why Python is so good.

Python Advantage

Python supports multiple programming paradigms primarily but not limited to object-oriented imperative and, to a lesser extent and functional programming styles. It features a dynamic type system and automatic memory management, similar to that of Ruby Pearl, and Tcl.

Python is often used as a programming language but is also used in a wide range of non-scripting contexts. Python code can be packaged into stand alone executable programs. Python interpreters are available for all operating systems.

Python Technology Advantage:

- Python is Open Source
- Python is Interpreted, high level languages
- Python is supported by large developer communities
- Python is easy to learn compared to C++, Java
- Python is easy to extend in C++, C, and Java
- Python runs on Windows, Linux/Unix, Mac OS , and has been ported to the Java and .NET virtual machines support for variable number of function arguments.

- Python can be used for scripting and general programming (CLI sapi, embedded etc)

Topics covered

Python Scripting Training Course Content

Faculty: Real time and certified

(Includes theoretical as well as practical sessions)

Chapter 1: An Introduction to Python

Introductory Remarks about Python

Strengths and Weaknesses

A Brief History of Python

Python Versions

Installing Python

Environment Variables

Executing Python from the Command Line

IDLE

Editing Python Files

Getting Help

Dynamic Types

Python Reserved Words

Naming Conventions

Chapter 2: Basic Python Syntax

Introduction

Basic Syntax

Comments

String Values

String Operations

The format Method



String Slices

String Operators

Numeric Data Types

Conversions

Simple Input and Output

The print Function

Chapter 3: Language Components

Introduction

Control Flow and Syntax

Indenting

The if Statement

Relational Operators

Logical Operators

True or False

Bit Wise

Operators The

while Loop

break and

continue The

for Loop

Chapter 4: Collections

Introduction

Lists

Tuples

Sets

Dictionaries

Sorting Dictionaries

Copying Collections



Summary

Chapter 5: Functions

Introduction

Defining Your Own Functions

Parameters

Function Documentation

Keyword and Optional Parameters

Passing Collections to a Function

Variable Number of Arguments

Scope

Functions - "First Class Citizens"

Passing Functions to a Function

Mapping Functions in a Dictionary

Lambda

Chapter 6: Modules

Standard Modules - sys

Standard Modules - math

Standard Modules - time

The dir Function

Chapter 7: Exceptions

Errors

Run Time Errors

The Exception Model

Exception Hierarchy

Handling Multiple

Exceptions raise

assert



Writing Your Own Exception Classes

Chapter 8: Input and Output

Introduction

Data Streams

Creating Your Own Data Streams

Access Modes

Writing Data to a File

Reading Data From a File

Additional File Methods

Using Pipes as Data Streams

Handling IO Exceptions

Working with Directories

Metadata

The pickle Module

Chapter 9: Classes in Python

Classes in Python

Principles of Object Orientation

Creating Classes

Instance Methods

File Organization

Special Methods

Class Variables

Inheritance

Polymorphism

Type Identification

Custom Exception Classes

Class Documentation – pydoc



Chapter 10: Regular Expressions

Introduction

Simple Character Matches

Special Characters

Character Classes

Quantifiers

The Dot Character

Greedy Matches

Grouping

Matching at Beginning or End

Match Objects

Substituting

Splitting a String

Compiling Regular Expressions

Flags



Why Python @ IGEEKS

- IGEEKS has an experienced team of Python experts.
- IGEEKS has delivered number of Workshop on Python in Top Engineering colleges.

Who can take up?

Teachers, Research Scholars UG and PG students from AICTE approved Engineering Colleges with relevant background.