Module 1:

Introduction

By Ninad Gaikwad

Reference - “Core Python Programming”
Dr. R. Nageshwara Rao
Dreamtech Press

Introduction

Developed by Guido Van Rossum in 1999

Features:

Simple	Dynamically Typed	Interpreted	Scripting Language
Easy to Learn	Platform Independent	Extensible	Database Connectivity
Open Source	Portable	Embeddable	Scalable
High Level Language	Procedure and Object Oriented	Huge Library	Batteries Included

Introduction

- Execution of a Python program:

Source code =>> [Compiler] =>> Bytecode (.pyc) =>> [PVM] =>> Machine code =>> Computer

- Viewing the Byte code

  python -m dis first.py

- Flavors of Python (Types of compilers):

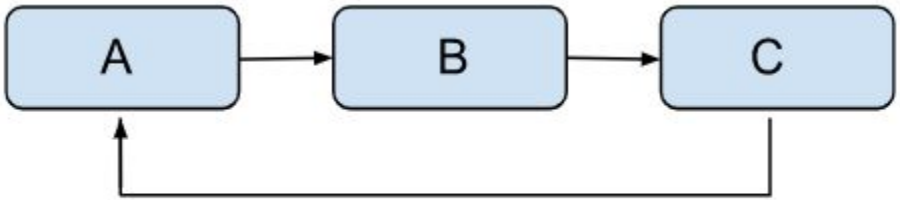
CPython	IronPython	RubyPython	Pythonxy
Jython	PyPy	StacklessPython	AnacondaPython

Introduction

- Python Virtual Machine (PVM):

Byte code =>> Python virtual machine (PVM / Interpreter) =>> Machine Code =>> Computer

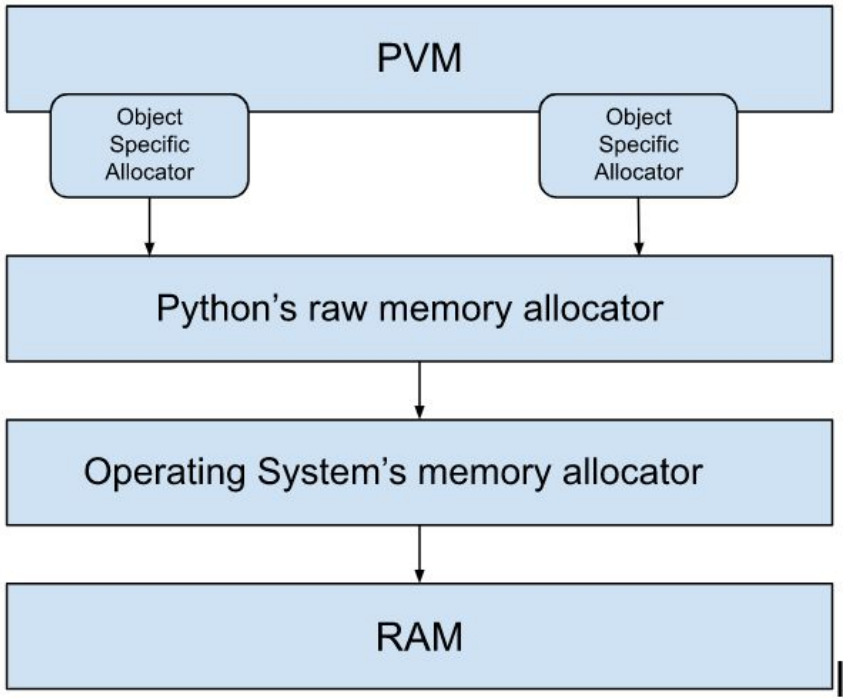
- Frozen Binaries
  - Eg .exe files
- Garbage collection in Python



Reference cycle of three objects

Introduction

- Memory management in python by PVM:



Comparison between C and Python

C	**Python**
Procedure Oriented	Object oriented
Faster	Slower
Declare Datatypes	Not required
Static and weak discipline	Dynamic and Strong
Pointers	Not available
No Exception Handling	Handles exception
Do..while, while and for	While and for
Switch case	Not available
Increment variable does not increment automatically	Loop variable increments automatically
Allocation, deallocation of memory manual	Automatic

Comparison between C and Python

C	**Python**
No garbage collector	present
Single and multi dimensional array	Multi dimensional supported by using numpy module
Array index positive	Can be negative
Checking allocation outside array allocation not possible	Possible
Indentation not necessary `	Indentation represents block of statements
Semicolon to terminate statement comma to separate expression	New line indicates end of statement comma is expression separator

Comparison between java and python

Java	**Python**
Object oriented, functional features added in java 8 through lambda expressions	Blends functional with object oriented features
Programs are verbose (Contain more number of lines)	Concise and compact
Datatype declaration compulsory	Not required
Type discipline is static and weak	Dynamic and strong
Do…while, while and for loops	While and for loops
Has switch statement	No switch statement
Variables in for loop not incremented automatically	For loop variable incremented automatically
Memory allocation done automatically by JVM	Memory allocation done automatically by PVM

Comparison between java and python

Java	**Python**
Supports multidimensional array	Multidimensional array can be implemented using numpy
Array index should be positive	Can be negative
Checking location outside array allocation not supported	Supported
Indentation not necessary	Indentation is required
Semicolon used to end statement, comma to separate expressions	New line used as end of statement, semicolon for separation of expression
Collection of objects like linked list or vector store only objects but not primary datatypes like integers	Collection objects can store objects of any type.
Supports multidimensional array	Multidimensional array can be implemented using numpy
Array index should be positive	Can be negative

Environment

- Installing Modules using python installation of packages (pip)
  - Numpy - Numerical python
  - Pandas - Used in data analysis
  - Xlrd - Retrieve data from microsoft excel spreadsheet files
  - Matplotlib - Produce good quality 2D graphics
  - Verify installed packages =>> help(‘modules’)
- Writing and executing program
  - Python’s command line window
  - Python’s IDLE(Integrated Development Environment) graphics window
  - Directly from system prompt
- Getting help in python
  - help()
  - help(‘print’)
  - Python documentation