Python is an [**interpreted**](https://en.wikipedia.org/wiki/Interpreted_language), [**high-level**](https://en.wikipedia.org/wiki/High-level_programming_language), [**general-purpose**](https://en.wikipedia.org/wiki/General-purpose_programming_language) [programming language](https://en.wikipedia.org/wiki/Programming_language).

Created by [**Guido van Rossum**](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991,

Other features Includes:

* [Dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)).
* Supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm),
  + [Procedural](https://en.wikipedia.org/wiki/Procedural_programming)
  + Object-Oriented and
  + [functional programming](https://en.wikipedia.org/wiki/Functional_programming).

Python Release:

The implementation of Python was started in the December 1989 by **Guido Van Rossum** at CWI in Netherland.

In February 1991, van Rossum published the code (labelled version 0.9.0)

Python 1.0 was released on January 1994. With new features like:

* lambda, map, filter, and reduce.

Python 2.0 was released on 16 October 2000. With new features.

* list comprehensions
* [Cycle-detecting](https://en.wikipedia.org/wiki/Cycle_detection) [garbage collector](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) and
* support for [Unicode](https://en.wikipedia.org/wiki/Unicode)

Python 3.0 was released on 3 December 2008. Major revision of the language includes

* 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python.

Python Latest Release:

* 2.x – 2.7.16 March 4, 2019
* 3.x – 3.7.4 July, 2019

Program to Process

# include <stdio.h>

# include <stdlib.h> b

int x = 20;

int main(){

int a = 10;

int \*ptr;

while(1){

ptr = (int\*) malloc(sizeof(int) \* 10);

free(ptr);

}

return 0;

}

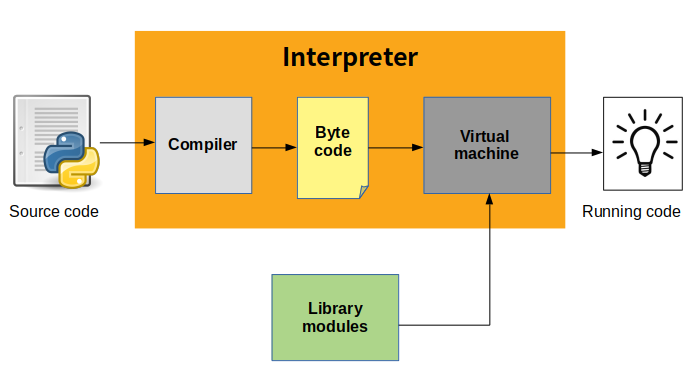


Notes:

C does not provide garbage collection (run time memory management) feature. So, user(developer) must take responsibility. Otherwise it creates memory leaks.

Python is having a VM which monitors the memory allocation and deallocation to private heap. So, no memory leaks

Python is Interpreted language?



# Python Basic

**Syntax:**

Based on Indentation

If True:

Line1

Line2

If True:

Line1

Line2

If True:

Line1

Line2

**Comments:**

One-line Comment:

**#** This is a comment

**#** Comment Line 2

Multi Line Comment:

**“””**

Multi Line comment

Line 2

**“””**

**Variables:**

**[Alphabet]+[Alphanumeric]+**

Name, empName, Ename, EmpName,

Name, emp\_name, ename, \_\_name

**123name, 123, adad#$#%**

**Keywords:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| False | class | finally | is | return |
| None | continue | for | lambda | try |
| True | def | from | nonlocal | while |
| And | del | global | not | with |
| As | elif | if | or | yield |
| Assert | else | import | pass |  |
| Break | except | in | raise |  |

**Operators:**

Python Arithmetic Operator **(+, -, \*, /, //, %)**

Python Relational Operator **(>, >=, <, <=, ==, != )**

Python Assignment Operator **(=, +=, -=, \*=, /=, //=, %=)**

Python Logical Operator **(and, or, not)**

Python Membership Operator **(in, not in)**

Python Identity Operator **(is, is not)**

Python Bitwise Operator **(&, |, ~, ^, >>, <<)**

**Data type:**

**Numeric**

Integer : a = 20 int

Floating-Point : a = 10.5 float

Complex : a = 1j complex

**Text**

String : a = “Apple” str

**Boolean**

Bool : a = True bool

**Sequence:**

List : a = [10, 20, 30] list

Tuple : a = (10, 20, 30) tuple

Range : a = range(10, 40, 10) range

**Mapped:**

Dict : a = {‘A’ : “Apple” } dict

**Set:**

Set : a = {“Apple”, “Boy”} set

Frozen Set : a = frozenset( {“Cat”, “Egg”}) frozenset

**Binary:**

Byte : a = b’Apple’ byte

Byte Array : a = bytearray(5) bytearray

**Conditions IF ELSE:**

a = 200  
b = 33  
if a > b:  
 print("A > B")

elif a < b:

print("A < B")  
else:  
   print("A == B")

print(“A > B”) if a > b else print(“A < B”) if a < b else print(“A == B”)

**Loops:**

**While**

while **CONDITIONS**: # **CONDITIONS(True/ False)**

Statements

**For**

for **VAR** in **SEQUENCE**: # **SEQUENCE(List/ Tuple/ Range)**

Statements

**Break Continue Pass**

**Number:**

**Integer <class ‘int’>**

a = 10 <class ‘int’> int()

b = 0b1010 <class ‘int’> b = bin(10) <class ‘str’>

x = 0xa <class ‘int’> x = hex(10) <class ‘str’>

o = 0o12 <class ‘int’> o = oct(10) <class ‘srr’>

**Floats**

a = 10.5 <class ‘float’>

b = 10.4e-04 <class ‘float’>

**Complex**

a = 2 + 3j <class ‘complex’>

**Strings:**

**Single Quotes:**

a = “Apple” or ‘Apple’

b = ‘This is an Apple’ \

‘ This is a Banana’

**Multi Quotes:**

a = “””

This is a Multi-Line string.

Line 1

Line 2

“””

**Raw String**

rs = r’This is \t a \nraw String’

**Formatted String**

a = 3

b = 100

s1 = f’{a} kg Apples at {b} = {a \* b}’

print(s1) #‘3 kg Apple at 100 = 300’

s2 = “%d kg Apple at %d = %d” % (a, b, a \* b)

print(s2) #‘3 kg Apple at 100 = 300’

s3 = ’{} kg Apples at {} = {}’.format(a, b, a \* b}

print(s3) #‘3 kg Apple at 100 = 300’

**String Operation:**

**Q & A:-**

1. [Python Program to Swap Two Variables](https://www.programiz.com/python-programming/examples/swap-variables)
2. [Python Program to Generate a Random Number](https://www.programiz.com/python-programming/examples/random-number)
3. [Python Program to Check if a Number is Odd or Even](https://www.programiz.com/python-programming/examples/odd-even)
4. [Python Program to Find ASCII Value of Character](https://www.programiz.com/python-programming/examples/ascii-character)
5. [Python Program to Display the multiplication Table](https://www.programiz.com/python-programming/examples/multiplication-table)
6. [Python Program to Check if a Number is Positive, Negative or 0](https://www.programiz.com/python-programming/examples/positive-negative-zero)
7. **Python Program to Print Pattern**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**1 1 1 1 1**

**2 2 2 2**

**3 3 3**

**4 4**

**5**

**Demo**

**Lists:**

**Tuples:**

**Sets:**

**Dictionary:**

**Functions:**

**Lambda:**

**Modules:**