

1. Language Fundamentals

1. Introduction

- Python is a general purpose high level programming language.
- Python was developed by Guido Van Rossum in 1989 while working at National Research Institute at Netherlands.
- But officially Python was made available to public in 1991. The official Date of Birth for Python is : Feb 20th 1991.
- Python is recommended as first programming language for beginners.

Eg1: To print HelloWorld:

```
In [1]: print("Hello World")
```

Hello World

Eg2: To print the sum of 2 numbers

```
In [2]: a=10
b=20
print("The Sum:", (a+b))
```

The Sum: 30

The name Python was selected from the TV Show "The Complete Monty Python's Circus", which was broadcasted in BBC from 1969 to 1974.

Guido developed Python language by taking almost all programming features from different languages

1. Functional Programming Features from C
2. Object Oriented Programming Features from C++
3. Scripting Language Features from Perl and Shell Script
4. Modular Programming Features from Modula-3

Most of syntax in Python Derived from C and ABC languages.

Where we can use Python:

We can use everywhere. The most common important application areas are :

1. For developing Desktop Applications
2. For developing web Applications
3. For developing database Applications
4. For Network Programming
5. For developing games
6. For Data Analysis Applications
7. For Machine Learning
8. For developing Artificial Intelligence Applications
9. For IOT
- ...

Note:

Internally Google and Youtube use Python coding
NASA and Nework Stock Exchange Applications developed by Python.
Top Software companies like Google, Microsoft, IBM, Yahoo using Python.

Features of Python:

1. Simple and easy to learn:

Python is a simple programming language. When we read Python program,we can feel like reading english statements.
The syntaxes are very simple and only 30+ kerywords are available.
When compared with other languages, we can write programs with very less number of lines. Hence more readability and simplicity.
We can reduce development and cost of the project.

2. Freeware and Open Source:

We can use Python software without any licence and it is freeware.
Its source code is open,so that we can we can customize based on our requirement.
Eg: Jython is customized version of Python to work with Java Applications.

3. High Level Programming language:

Python is high level programming language and hence it is programmer friendly language.
Being a programmer we are not required to concentrate low level activities like memory management and security etc..

4. Platform Independent:

Once we write a Python program,it can run on any platform without rewriting once again.
Internally PVM is responsible to convert into machine understandable form.

5. Portability:

Python programs are portable. ie we can migrate from one platform to another platform very easily. Python programs will provide same results on any paltform.

6. Dynamically Typed:

In Python we are not required to declare type for variables. Whenever we are assigning the value, based on value, type will be allocated automatically.Hence Python is considered as dynamically typed language.
But Java, C etc are Statically Typed Languages b'z we have to provide type at the beginning only.
This dynamic typing nature will provide more flexibility to the programmer.

7. Both Procedure Oriented and Object Oriented:

Python language supports both Procedure oriented (like C, pascal etc) and object oriented (like C++,Java) features. Hence we can get benefits of both like security and reusability etc

8. Interpreted:

We are not required to compile Python programs explicitly. Internally Python interpreter will take care that compilation. If compilation fails interpreter raised syntax errors. Once compilation success then PVM (Python Virtual Machine) is responsible to execute.

9. Extensible:

We can use other language programs in Python.
The main advantages of this approach are:

1. We can use already existing legacy non-Python code
2. We can improve performance of the application

10. Embedded:

We can use Python programs in any other language programs.
i.e we can embedd Python programs anywhere.

11. Extensive Library:

Python has a rich inbuilt library.
Being a programmer we can use this library directly and we are not responsible to implement the functionality.
etc...

Limitations of Python:

1. Performance wise not up to the mark b'z it is interpreted language.
2. Not using for mobile Applications

Flavors of Python:

1.CPython:

It is the standard flavor of Python. It can be used to work with C lanugage Applications

2. Jython or JPython:

It is for Java Applications. It can run on JVM

3. IronPython:

It is for C#.Net platform

4.PyPy:

The main advantage of PyPy is performance will be improved because JIT compiler is available inside PVM.

5.RubyPython

For Ruby Platforms

6. AnacondaPython
It is specially designed for handling large volume of data processing.
...

Python Versions:

Python 1.0V introduced in Jan 1994
Python 2.0V introduced in October 2000
Python 3.0V introduced in December 2008
Note: Python 3 won't provide backward compatibility to Python2
i.e there is no guarantee that Python2 programs will run in Python3.
Current versions
Python 3.6.1 Python 2.7.13

2. Identifiers

A name in Python program is called identifier.
It can be class name or function name or module name or variable name.
`a = 10`

Rules to define identifiers in Python:

1. The only allowed characters in Python are

- alphabet symbols(either lower case or upper case)
 - digits(0 to 9)
 - underscore symbol(_)
- By mistake if we are using any other symbol like \$ then we will get syntax error.
Ex:-
- `cash = 10` ✓
 - `ca$h =20` x

2. Identifier should not starts with digit

- `123total` x
- `total123` ✓

3. Identifiers are case sensitive. Of course Python language is case sensitive language.

- `total=10`
- `TOTAL=999`
- `print(total)` #10
- `print(TOTAL)` #999

Identifier:

1. Alphabet Symbols (Either Upper case OR Lower case)
2. If Identifier is start with Underscore (_) then it indicates it is private.
3. Identifier should not start with Digits.
4. Identifiers are case sensitive.
5. We cannot use reserved words as identifiers
Eg: `def=10` x
6. There is no length limit for Python identifiers. But not recommended to use too lengthy

identifiers.

7. Dollor (\$) Symbol is not allowed in Python.

Q. Which of the following are valid Python identifiers?

- 1) 123total x
- 2) total123 ✓
- 3) java2share ✓
- 4) ca\$h x
- 5) _abc_abc_ ✓
- 6) def x
- 7) if x

Note:

- 1. If identifier starts with _ symbol then it indicates that it is private
- 2. If identifier starts with __ (two under score symbols) indicating that strongly private identifier.
- 3. If the identifier starts and ends with two underscore symbols then the identifier is language defined special name, which is also known as magic methods.
Eg: __add__

3. Reserved Words

In Python some words are reserved to represent some meaning or functionality. Such type of words are called Reserved words.

There are 33 reserved words available in Python.

- True, False, None
- and, or, not, is
- if, elif, else
- while, for, break, continue, return, in, yield
- try, except, finally, raise, assert
- import, from, as, class, def, pass, global, nonlocal, lambda, del, with

Note:

- 1. All Reserved words in Python contain only alphabet symbols.
- 2. Except the following 3 reserved words, all contain only lower case alphabet symbols.

- True
- False
- None

Eg: a= true x
a=True ✓

```
In [4]: import keyword
```

```
In [7]: print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

