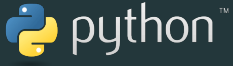
Programming in Python

Python 101 : Introduction



Hemant, K. // hemantk // hemantk@sg.ibm.com

April 8, 2020

IBM Systems | Singapore

Table of Contents

URL: https://www.python.org/ @ThePSF

Identity, Trademarks and Images

Python is an open source programming language governed under Python Software Foundation (PSF). The relevant images and terms are PSF trademark and used only for educational purposes.

Disclaimer

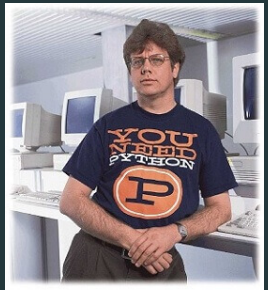
The presented material is written and compiled by author from various sources, and wherever certain items are taken as-it-is following Creative-Common-License framework. The original citation is given with proper attribution. The contents are mostly based on Python Software Foundation (PSF) web page and in case of any mistake, the reference at PSF will prevail to be authentic.

1. Python as a programming language

History of Programming Languages and Evolution of Python

- First developed by Guido van Rossum at National Research Institute for Mathematics and Computer Science (CWI) in Amsterdam. Later some 20 odd developers joined.
- Named after BBC Channel 1 famous comedy series "Monty Python Flying Circus" aired during 1969-74. The 6 main characters were famously called the Pythons.
- The CWI team has actively working with a language based on ALGOL and created **ABC**.
- Guido worked on Unix function `glob()` with Lambert Meertens, and which was used in ABC and also with **Amoeba project** with Andrew Tenebaum for UNIX kernel as MINIX.
- Guido has been using C, Awk, Tcl/Tk and recently invented multi-purpose language Perl, but mainly scripting languages.
- This lead his interest to work during Christmas of 1990's, and created a interpreted language based on C, ABC, Awk, Perl and named in after "the Pythons" as **python** and evolved as *Language of Computer Programming for Everybody*.

Is Python interpreted or compiled language? What is interpreted language?



Centrum Wiskunde en Informatica (CWI) https://www.cwi.nl/

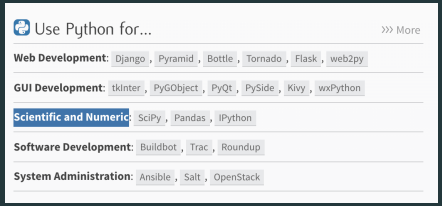
Python: Traditional Programming vs Scripting

Traditional Programming or System Programming (Fortran, C, C++, Java, *etc.*)

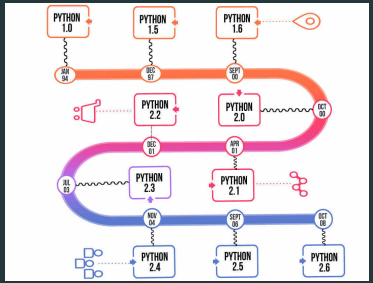
- building (usually large, monolithic) applications
- takes longer time to write
- very fast

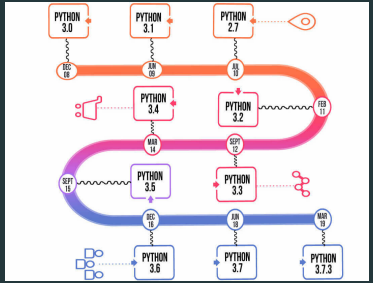
Scripting (Python, Perl, MATLAB**, *etc.*)**

- scripting means programming at a high and flexible abstraction level
- simple and clean syntax of the command languages
- tight integration of simulation and visualization
- this style is better if the most important resource is programmer's time



Evolution of Python : python 2.x vs python 3.x





e.g.

»» print ”Hello World!”

»» 2/3

0

»» 3/2

1

e.g.

»» print(”Hello World!”)

»» 2/3

0.6666666666666666

»» 3/2

1.5

Some minor differences in python2 and python3 versions are: function print and integer divisons. lib **2to3** helps port py2 codes in py3. For more refer to: https://www.guru99.com/python-2-vs-python-3.html

Courtsey: Infographic images from www.Geeks4Geeks.com

2. Setting the environment

Evolution of Python : python2 to python3

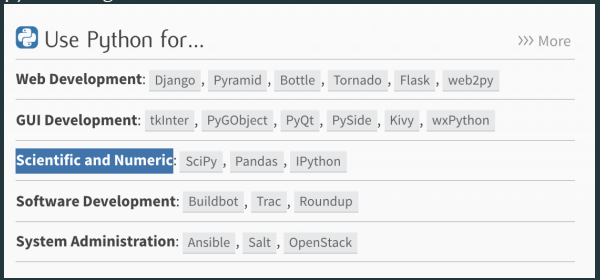
- $ python2

```
> python2

WARNING: Python 2.7 is not recommended.
This version is included in macOS for compatibility with legacy software.
Future versions of macOS will not include Python 2.7.
Instead, it is recommended that you transition to using 'python3' from within Terminal.

Python 2.7.16 (default, Feb 29 2020, 01:55:37)
[GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.29.20) (-macos10.15-objc- on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

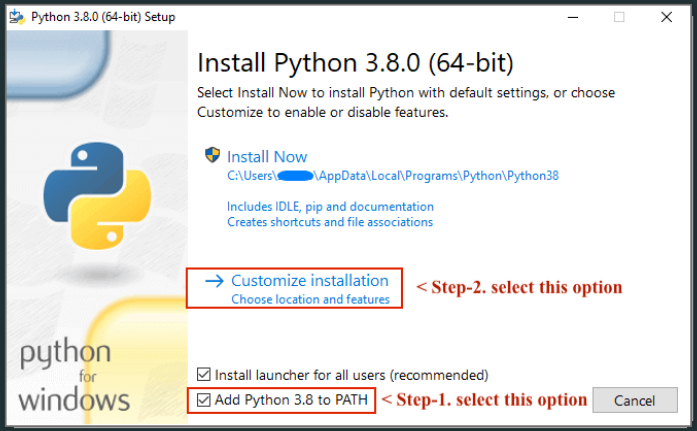
- $ python3

```
> python3
Python 3.7.6 (default, Dec 30 2019, 19:38:26)
[Clang 11.0.0 (clang-1100.0.33.16)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- python usage



Python Installation on Windows



Environment Settings and Getting Started wit Python

- Download Python from www.python.org/download
- PyPI : Python Package Installer

```
$ pip install pylint
```

- Checking if a package is installed

```
$ python -c "import math"
$ echo $?
 0     # return '0' mmeans math module exists in system
```

- Checking if a package is installed

```
$ python -c "import numpy"
$ echo $?
 1     # return '1' means numpy module does not exists in system
$ pip install numpy
```

```
$ python -c "import math"
$ echo $?
0                                  # math module exists in system

$ python -c "import numpy"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named numpy
$ echo $?
1                                  # numpy module does not exist in system
```

3. Learning Path

Learning Modules

Module # py101.1

- Python atomic data types
- Python data structures
- Conditional statements and loops
- Comprehensions
- Functions

Module # py101.2

- NumPy
- Python for data analysis using pandas

Module # py101.3

- Object-oriented programming (use of classes, objects, etc.)
- Data visualization (matplotlib, ggplot)
- Generator expressions

Hands On Exercise on jupyter notebook

1. If yiou are using Python Software Foundation shipped Python, use

```
pip3 install jupyter
```

2. If you are using Continuum Software Anaconda version of Python

```
conda install jupyter
```

To launch the notebook, issue the following command

```
jupyter notebook
```

References

1. PyPI: Python Package Index
   https://pypi.org/
2. PyCham IDE
   https://www.jetbrains.com/pycharm/
3. The History of Programming Languages
   https://visual.ly/community/Infographics/computers/history-programming-languages
4. Interview of Guido van Rossum: “I had rather write code than papers”
   https://www.cwi.nl/news/blogs/interview-guido-van-rossum-201cid-rather-write-code-than-papers.201d

Thank You !

for your patience ☺

hemant, k.