1
Introduction of Python

“Python is an experiment in how much freedom programmers need, too much freedom and nobody can read any other’s code. Too little and expressiveness endangered”
~Guido Van Rossum

Topics covered in this Chapter

1.1 Why Python

Python is a very popular and widely used general purpose interpreted programming language. Its is a dynamic type system and automatic memory management with multiple programming paradigms like imperative functional and procedural. Programs can be created to automate various task. Now a days all major IT companies like Google, Microsoft, Apple, etc, are using python as main programming language. Python is easiest programming language to learn quickly in short period of time. It can develop any type of application such as Desktop, Web application, Mobile application, utilities tools, with help of wide range of framework and libraries availability with minimum coding. It is free and open source with proper documentation available over internet. Python basically focuses on business logic instead of basic facts of programming. Python is available in various versions but primarily we will focus on 3.7 and upcoming is Python 3.8 which will release on October 2019. This will be full of new features. Python now focusing on performance related issues. Python is dynamically typed, automatic garbage-collected and memory management with multiple programming paradigms like procedural, OOP (Object Oriented Programming), functional etc. Python is agile language that allows us to do so many customized things with in small portion of time. And we can start application development from any point of module and in the ending phase of application development, can be join each other just before releasing

1.1.1 Drawbacks of Python

All programming languages have some advantages and disadvantages too. Python also have some disadvantage some of them are given below

1) Python is very slow in terms of execution of program in comparing other programming languages such as C, C++, Java. Java is the fastest language because of JVM (Java Virtual Machine) and JIT(Just In Time) compiler. Further information Refer to YouTube videos.
2) Drawing sophisticated graphics is computationally quite expensive. This causes reduces graphics quality.
3) Without Cython code execution is very slow. Cython enables you to compile your code at C level. C compiler optimize the things.
4) GPU or Cython is required for better performance.
5) We can limit speed of the application in order to wast resources.

1.2 History of Python

Python isn’t named after the type of snake. it’s actually named after a British Comedy group called “Monty Python’s Flying Circus.” Python language was created by “Guido van Rossum” in 1991 as 1.0, after 10 years its 2.0 was released in year 2000 with new features like comprehensions, cycle detecting garbage collection, and Unicode support. He was a big fan of of the group and their quirky humor. Python programs often the group of jokes and famous quotes in their code as tribute Because of various major updates python is available in two major versions Python 3.x and 2.x. python 2.x is a legacy version that will continue up to 2020 while python 3.x is fast updated and popular version. Some features can be import from python 3 using **__future__** in python 2.x. Python 3.0 was released in year 2008 and it was the major release without backward compatibility, it means python 2.x code can not run over 3.x series, this major issue was fixed in python 2.7 version because large amount of code was written in 2.X series. Initially this support was up to 2015 but now it has been extended up to 2020. Mr Guido took this responsibility up to July 2018, now he shares his leadership of five-person steering council. Now steering council will release next upcoming versions.

1.3 Major features of Python

- ◆ Easy to learn and reading because of the tabular space instead of curly braces.
- ◆ Because of using English language python programs becomes so easy to code
- ◆ Python is based on Interpreted language which saves lot of time to execute out application.
- ◆ Procedural programming, Object Oriented programming and functional programming support using Thread module python supports Multi Threading
- ◆ Python Android and iOS Templates parsing to ease generating dynamic page for client
- ◆ Interactive language, python code never converts human readable code for execution. This feature makes python very interactive by making changes at run-time.
- ◆ Python Android and iOS support by KIVY framework
- ◆ Platform independent same code can be executed on all available platforms.

1.3.1 What is new in upcoming Python 3.8

- **Assignment Expression** is known as *walrus operator* (:=). It is like eyes ans tusk of walrus. It allows value to assigned to a variable even a variable which is doesn’t exists. It follows list comprehension and lambda function tradition.

```
input_text=input("Please input text> ")
while input_text!="quit":
        print("You have entered ", input_text)
        input_text=input("Please input data> ")
```

and the above given program can be written like this

```
while (input_text := input("> ")) != "quit":
        print("You entered:", input_text)
```

- **Positional Only Parameters** helps us to remove ambiguity about which argument in a function definition position and which are keyword arguments. You can achieve it with the help of forward slash (/) and star (*) where (/) indicates some function parameters must be specified prepositionally and can’t be used as keyword argument. You can define functions positional only parameters. Here developer can force some arguments to be positional only. Lets see in the given example

```
>>def myfunction (a,b,/,c,d,e,*,f,g,h):
>>…  print(a,b,c,d,e,f,g,h)
>>myfunction(1,2,3,4,5,6,7,8)
Traceback (most recent call last):F
    File "<pyshell#15>", line 1, in <mmyfunction(1,2,3,4,5,6,7,8)
    TypeError: myfunction() takes 6 positional arguments but 8 were given
```

in the given example

- **Python Initialization and Configuration** Python 3.8 adds a new C API to configure the Python initialization providing better control on the whole configuration and better error reporting. Python code configuration is scattered all around the code. It helps to read and modify configuration before it is applied.
- **Vectorcall** is a fastest calling protocol for Cpython. It is the default implementation of python, it compiles the source code into intermediate bytecode, which is executed by the Cpython virtual machine. It allows for far faster to internal Python methods without the overhead of creating temporary objects to handle the call. Any extension type implementation a callable can use this protocol.
- **Run-time audit hooks** provide two API in the Python run-time for hooking events and making them observable to outside tools like testing frameworks or logging and auditing systems. An audit hook is an exit point that allows the auditor to add the modules subsequently. This happens by activating the hook to transfer control to an audit module.

The verified open hook allows python administrator and embedders to integrate with operating system support when launching scripts or importing python code. Audit hook and a verified open hook allow applications and frameworks written in pure python code to take advantage of extra notifications.

- **Pickle protocol 5 without of band data buffers** helps us to transfer compatible data separately from the main pickle stream, at the discretion of the communication layer. The process in which a python object to byte stream known as “pickling” or “serialization or marshaling. When pickle is used to transfer large data transfer between the python process in order to take full advantage of multi-core. Its important to optimize the transfer by reducing memory copy.
- **Typing Related hints** Python 3.8 is more mature now with release of some new features like literal types, typed dictionaries, final objects, protocols, etc. Typing defines a standard notation for python function and variable type notations. The notation can be used for documenting code in a concise, standard format, and it has been designed to also be used-by static and run-time type checker, static snalyzer, IDE and other tools. Python supports type hints, in the given example number should be a float and double() function should return a float, as well. Python treats these annotation as hints. They are not enforced at run-time. Here acceptFloat() function accepts string too as an argument.

```python
def acceptFloat(number: float) -> float:
    return 2 * number

print(acceptFloat(2.45343))
print("This is not a float value")
```

```
=================output======================================
4.90686
This is not a float value
===========================================================
```

There are several static type checkers are available like Pyright, Pytype, Pyre and MyPy which comes along with PyPi.

- **Parallel file system cache for compiled byte-code** Python 3.8 introduces a new PYTHONPYCACHEPEFIX settings that helps to configure the implicit byte code cache to use a separate parallel file system. Cache of is reported in **sys.pycache_prefix** and **none**
- **Debug build shares ABI as release builds** Python 3.8 uses same ABI whether it is built in release or debug mode. Now its is possible to load C extension built using stable ABI. Python 3.8 release builds and debug build ABI compatible. On Unix, C extensions are no longer linked to libpython, except on Android and Cygwin. It is also possible to statically link python to old C extensions using a shared python library.
- **f-string support a handy (=) specifiere for debugging** It is a form of debugging where print statements are inserted to print value of expressions or variables that we need to track with wring logs. Logs are very useful in production environment. f-string was

introduced in python 3.6. you can evaluate an expression as part of the string along with inserting result of function calls.

```
name="Tarkeshwar Barua"
print(f"Hello, Mr. {name}")
print(f"Hello, Mr. {name.capitalize()}")
print(f"Hello, Mr. {name.upper()}")
print(f"Hello, Mr. {name.lower()}")
print(f"{name.upper() = :-^10}")
print(f"{name.upper() = :->10}")
print(f"name={name}")
print(f"{name=}")

===============================output===========================
Hello, Mr. Tarkeshwar Barua
Hello, Mr. Tarkeshwar barua
Hello, Mr. TARKESHWAR BARUA
Hello, Mr. tarkeshwar barua
name.upper() = TARKESHWAR BARUA
name.upper() = TARKESHWAR BARUA
name=Tarkeshwar Barua
name='Tarkeshwar Barua'
================================================================
```

- ***continue* is now legal in *finally* block** A continue statement can’t exit a finally block when a continue statement occurs within a finally block, the target of thee continue statement must be within the finally block else a compile-time error occurs. In polder version of python doesn’t allow continue in a finally block because all code after the continue will never be executed. It was because its interpretation would have be problematic.
- **The default *asyncio* event loop is now ProactorEventLoop in Windows** The asyncio stands for asynchronous input output and refers to a programming paradigm which achieve high concurrency using a single thread of event loop. The event loop is the place where most of the magic happens in asyncio. The asyncio module introduced in pyton3.4. It provides convenient shortcuts to accessing the methods of the global and default policy. Tornado requires the add_reader family of methods, so its is not compatible with the ProactorEventLoop on Windows. It is currently provides two implementations of event loops SelectorEventLoop and ProactorEventLoop. The ProactorEventLoop is incompatible with SSL and the default loop supports SSL on Windows. By extension aiohttp should also works with https on Windows. Event loop is based on the selectors module. Subclass of AbstractEventLoop uses the most efficient selector available on windows sockets are supported. Proactor event loop for windows uses IOCP.
- **The *spwan* start method is now is used by default in multiprocessing (Mac)** The multi-threading tests were failing on Linux for everything except Python3.4. Spawn is a

python package that allows user to concisely specify and execute a large number of tasks with complex and co-dependent input parameter variances. It is used where thousands of similar simulations with input parameter variations are run for design and certification purpose. It allows specification of such large sets to be formulated in a concise input file.

```
from multiprocessing import Pool
from os import getpid

def calculateSquare(i):
 print("Process No. ", getpid())
 return i ** 2

if __name__ == '__main__':
 with Pool() as pool:
 result = pool.map(calculateSquare, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15])
 print(result)
```

```
=======================================Output=========================
Process No.  21044
Process No.  21456
Process No.  21044
Process No.  21456
Process No.  21044
Process No.  21456
Process No.  21044
Process No.  21044
Process No.  21456
Process No.  5180
Process No.  21456
Process No.  5180
Process No.  21044
Process No.  21456
Process No.  5180
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225]
======================================================================
```

- ***multiprocessing*** **can now use shared memory segment to avoid pickling cost between process** allows regions of memory to be created and shared between different python processes. In earlier versions of python, data could be shared between processes only by writing it out to a file, sending it over a network socket, or serializing it using pickle module. Shared memory provides a faster path for passing data between processes, allowing python to more efficiently use multiple processors and processors core. Shared memory segments can be allocated as raw regions of bytes, or they can use immutable list-like objects that stores a small subset of python objects numeric types, strings, byte objects, and the None object

- **type_ast is merged back to Cpython** There is a fork of the ast module(in C language) named *typed_ast* used by mypy, pytype and IIRC also by some linters. Its redeeming quality is that preserves certain comments. it's hard work to keep this coe upto date with developments in the language's grammar.

- **by-default now pickle uses protocol 5 to improve performance.** Provides a new way to pickle object that implement python buffer protocol such as byte, memory views, and NumPy arrays. Pickle is used to transfer data between python processes in order to take advantage of multi-core or multi-machine processing, it is important to optimize the transfer by reducing memory copies and possibly by applying data-dependent compression. Pickle cuts down on the number of memory copies that have to be made for such objects.

The above given information is given based on publish papers by https://docs.python.org/3/

1.4 Market Demand

There so many programming languages are being used to develop application in the market. But I would like to introduce with some facts those are taken from various sources of internet. These facts are given as primary introductory language, Sector wise demand, salary wise, market share.

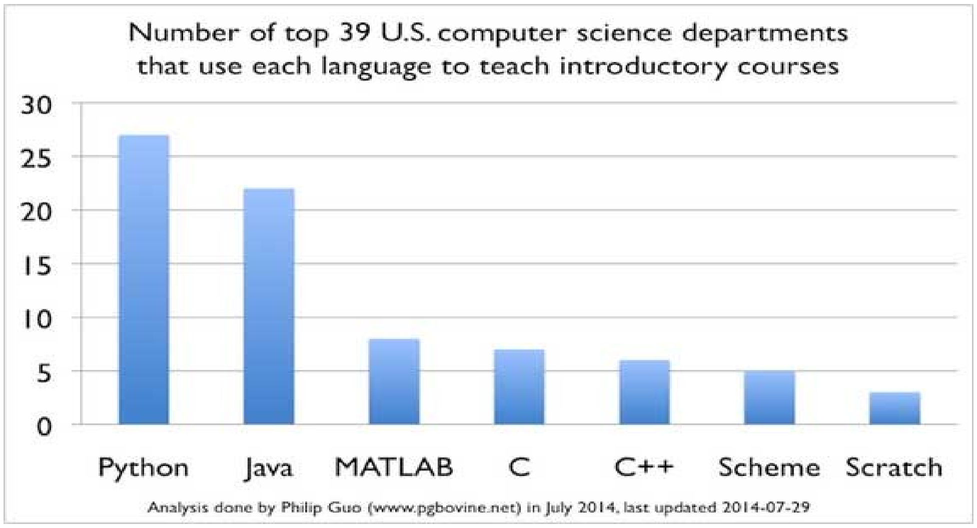
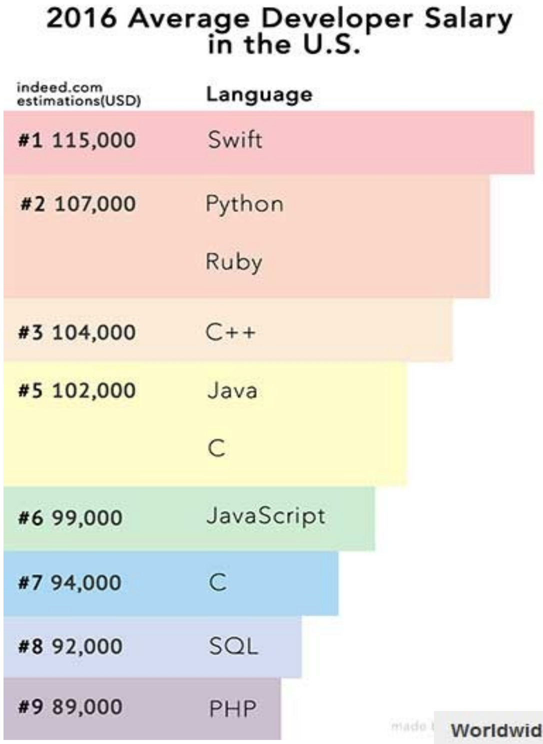


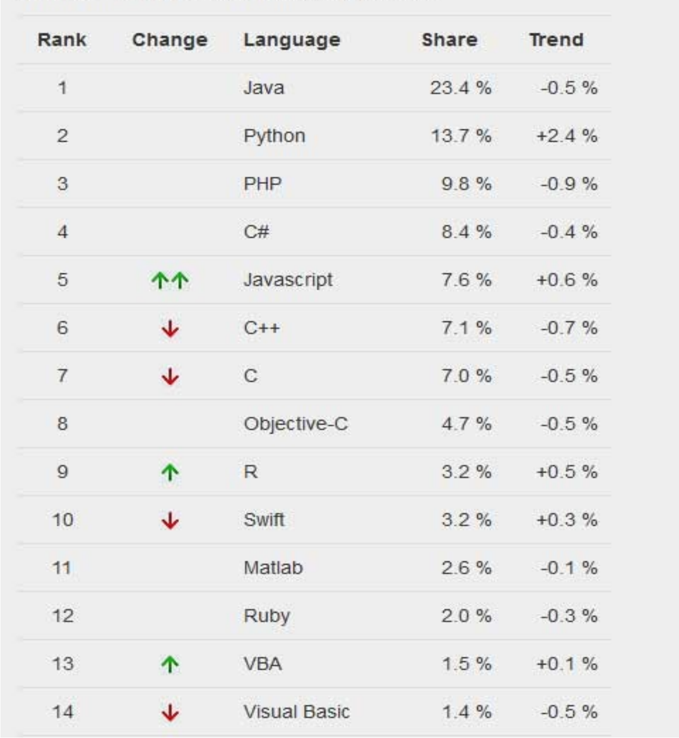
Figure 1.1 Market demand of programming language



2016 Average Developer Salary in the U.S.

indeed.com estimations(USD)	Language
#1 115,000	Swift
#2 107,000	Python Ruby
#3 104,000	C++
#5 102,000	Java C
#6 99,000	JavaScript
#7 94,000	C
#8 92,000	SQL
#9 89,000	PHP

Figure 1.2 Language wise average salary



Worldwide, Nov 2016 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Java	23.4 %	-0.5 %
2		Python	13.7 %	+2.4 %
3		PHP	9.8 %	-0.9 %
4		C#	8.4 %	-0.4 %
5	↑↑	Javascript	7.6 %	+0.6 %
6	↓	C++	7.1 %	-0.7 %
7	↓	C	7.0 %	-0.5 %
8		Objective-C	4.7 %	-0.5 %
9	↑	R	3.2 %	+0.5 %
10	↓	Swift	3.2 %	+0.3 %
11		Matlab	2.6 %	-0.1 %
12		Ruby	2.0 %	-0.3 %
13	↑	VBA	1.5 %	+0.1 %
14	↓	Visual Basic	1.4 %	-0.5 %

Figure 1.3 Language wise popularity

1.5 Why Python in Mobile app development?

Now a days mobile becomes very important part of our daily life therefore it is almost impossible for people to live without their mobile phones. Mobile phones has made our life so easy. By keeping this in our mind today most of the software companies have been moved to mobile app development but there are few challenges to developed mobile app due to various mobile phone platform like Android, iOS, Windows, etc. and all these platforms have different software requirements it means programmers need to write code for each natively and every platform which is very time consuming job therefore cross-platform of is good approach. Python is very easy and rich in library support that makes app development so easy. We can see in the following figure to market demand in various sectors which is enough to explain importance of python. Python is considered easy to learn and run almost anywhere. It provides a strong support for integration with several technologies and higher programming productivity across the development life cycle. It is good for large and complex projects with changing requirements. It is fastest growing language. It runs on several millions of phones including various industries. Python codes are more easy to read.

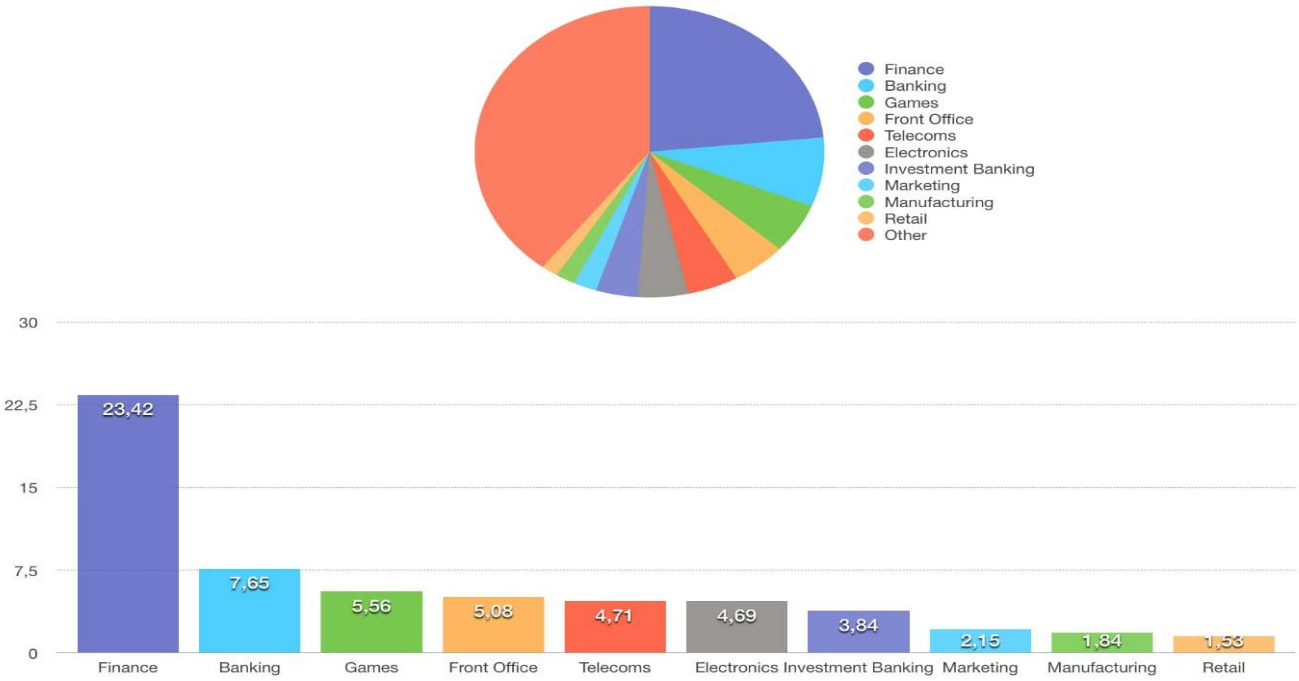


Figure 1.4 Industry serctor wise market demand

1.6 Python Versions.

Table 1.1 Python Versions

Version	**Release Date**
Python 1.0	1991
Python 2.0	Oct 2000
Python 2.1	April 2001
Python 2.2	Dec 2001
Python 2.3	July 2003

Python 2.4	Nov 2004
Python 2.5	Sep 2006
Python 2.6	Oct 2008
Python 2.7	July 2010
Python 3.0	Dec 2008
Python 3.1	June 2009
Python 3.2	Feb 2011
Python 3.3	Sep 2012
Python 3.4	March 2014
Python 3.5	Sep 2015
Python 3.6	Dec 2016
Python 3.7	June 2018
Python 3.7	October 2019

You can check you installed python version using following command at terminal/command prompt

```
$ python --version
      Or
$ python3 --version
```

1.7 Architecture of python application

Python syntax and style is flexible while it comes to structuring your application, there are so many opinion about its structure but here is the most commonly used structure by the programmers, It is very useful to managing and distributing application easily. Here is given some basic program structure. Where "*helloworld*" is your folder name/project name which contains all the required file.

.gitignore – It tells git which file need to ignore, like configuration files, IDE clutter

helloworld.py – This is the main program to execute which contains all the python code.

LICENSE – It is a plain text file which describes license of project, its mandatory to have while distributing code. This file name always in all caps letter

README.md – stands for Markdown file (project documentation file). This is very important to craft this file, you can create this file easily by this URL https://dbader.org/blog/write-a-great-readme-for-your-github-project

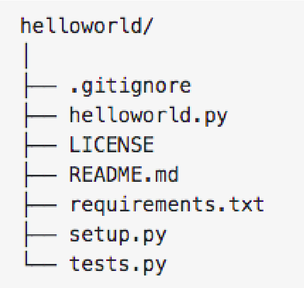
requirements.txt – Here you can specify all your program dependencies along with the its version. Adding dependency in this file called pinning, pip freeze command can be used to add all the production dependency in this file. It is useful only in development environment. Adding dependency into this file is called pinning. Pip and freeze commands are used to do so.

```
pip install -r < requirements.txt
```

setup.py - This is also doing the same job as requirements.txt file does but in package redistribution. This file useful in end-user system dependency installation, where –user indicate site-packages directory for running python.

```
python setup.py install --user
```

test.py - This is file is responsible for you tests, to make sure you application proper working.



```
helloworld/
│
├── .gitignore
├── helloworld.py
├── LICENSE
├── README.md
├── requirements.txt
├── setup.py
└── tests.py
```

Figure 1.5 A Python Program structure

1.8 System Requirement

All the given code in the this book are valid for both python version but for the best result tested environment with python 3.x which is the given below-

Table 1.2 System Requirement

Specification	**Description**
Python Version	2.x(2.3.1 to 2.7.16) and 3.x(3.3.0 to 3.7.3)
Processor	Pentium-4 or higher
RAM	512 MB(1 GB recommended)
Operating System	Windows XP (incl. 64Bit) or heigher, Linux(Any), Mac OS X 10.5, Raspberry Pi,
Screen Resolution	800*600

1.9 Installation of python in various platform

Installation in windows

after downloading python **.exe** package installer from https://www.python.org/, just double click on it and follow the instructions as its given click next and finally finish. you need to set windows environment path too temporary can be set from the command line command is given below

```
C:\Users\admin>set path= "C:\Program Files\python\bin\"
```

and permanent path can be set using environment variable by right click on My Computer → Properties → Advanced → Environmental Variable → System Variables → Path → Edit → add you python path after putting; (semi-colon), Then OK,

Installation in Linux

In Linux python installation may different according to various distribution but most popular distribution is Ubuntu, installation command is given below

```
sudo apt install python
Or
sudo apt install python3
```

for other distribution of Linux you can download Gzipped source tar file from the https://www.python.org/ and install it from Terminal and add python executable location has been added to the PATH environment variable. Python can be installed from the source code too, just your need to download it from internet. After unzip source code apply ./configure command and finally make command to generate executable add environment PATH

```
pip install -r < requirements.txt
```

```
wget --no-check-certificate https://www.python.org/ftp/python/2.7.X/Python-2.7.X.tgz tar -xzf python-2.7.X.tgz
cd Python-2.7.X
./configure
make
sudo make install
```

Installation in Mac

Mac comes with pre-installed python but it could be outdated you can download python **.pkg** package from https://www.python.org/, just double click on it and follow the instructions as its given click next and finally finish and add python executable location has been added to the PATH environment variable

1.10 Creating First Hello World App

You can write python script in any ASCII/ANSI based editor such as IDLE, Eclipse, Netbeans, Visual Studio, PyCharm, Notepad, Notepad++, Sublime, Jupiter Notebook, etc. you just require to download and install python interpreter according to your operating system from https://www.python.org/.

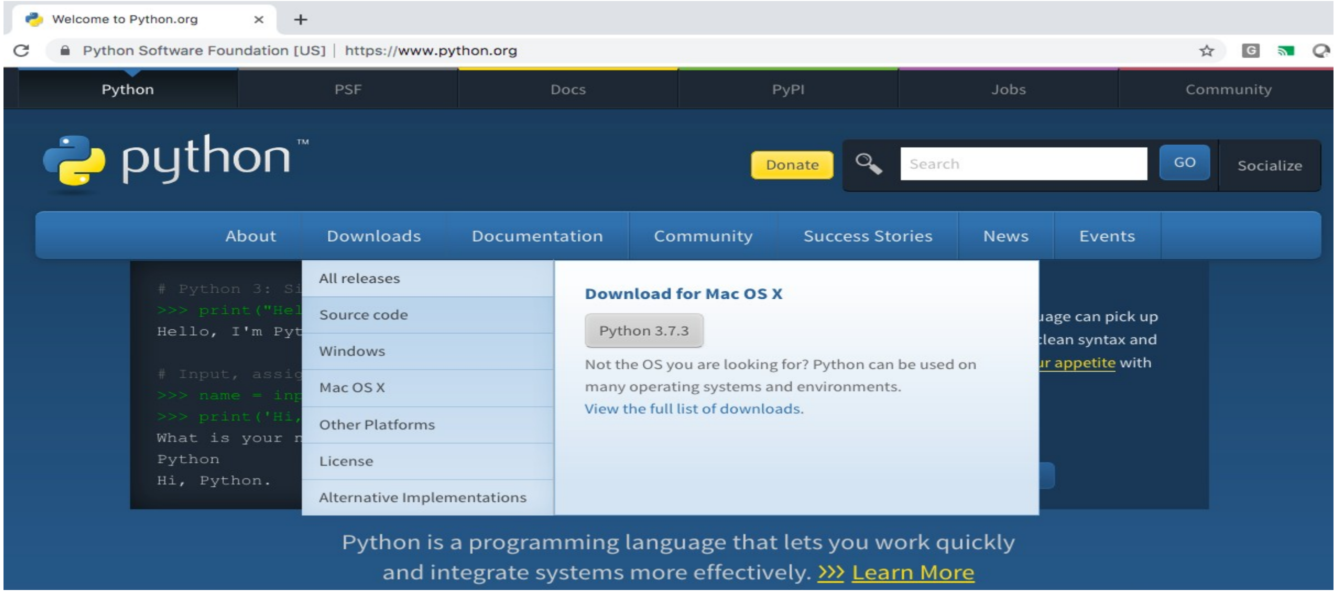
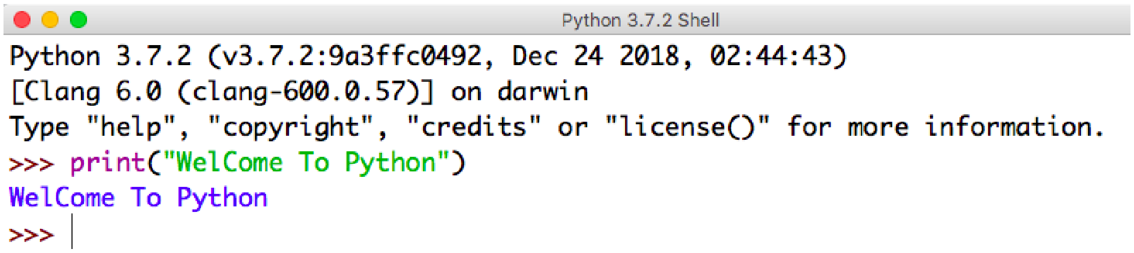

Welcome to Python.org
Python Software Foundation [US] | https://www.python.org
Python
PSF
Docs
PyPI
Jobs
Community
python™
Donate
Search
GO
Socialize
About
Downloads
Documentation
Community
Success Stories
News
Events
All releases
Source code
Windows
Mac OS X
Other Platforms
License
Alternative Implementations
Download for Mac OS X
Python 3.7.3
Not the OS you are looking for? Python can be used on many operating systems and environments.
View the full list of downloads.
Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Figure 1.6 https://www.python.org home page

After download and installation you can writing code built-in text editor IDLE, snapshot is given below

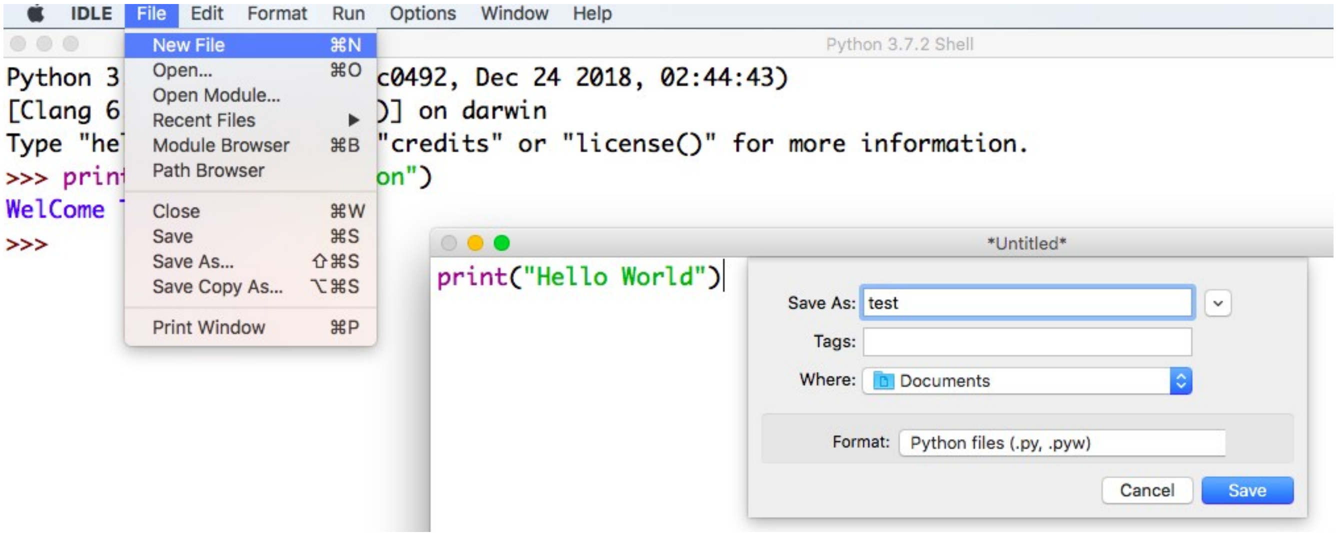


```
Python 3.7.2 (v3.7.2:9a3ffc0492, Dec 24 2018, 02:44:43)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("WelCome To Python")
WelCome To Python
>>> 
```

Figure 1.7 Python command line interpreter

Writing program in python is very simple, as we have seen in the given snapshot. Here >>> symbol indicates the line where you need to write your code/instruction and this code can be interpreted while you press enter button and it will display result immediately. These commands are sequential. You can save these code in your file with extension name **.py** as given in snapshot. You can close/exit from the python shell using exit/quit command as suggested below

```
quit()              #ctrl+D
          Or
exit()              #ctrl+C
          Or
# command+q in mac
```



```
Python 3                    c0492, Dec 24 2018, 02:44:43)
[Clang 6                    )] on darwin
Type "he                    "credits" or "license()" for more information.
>>> prin                    on")
WelCome
>>>
```

```
print("Hello World")
```

Save As: test

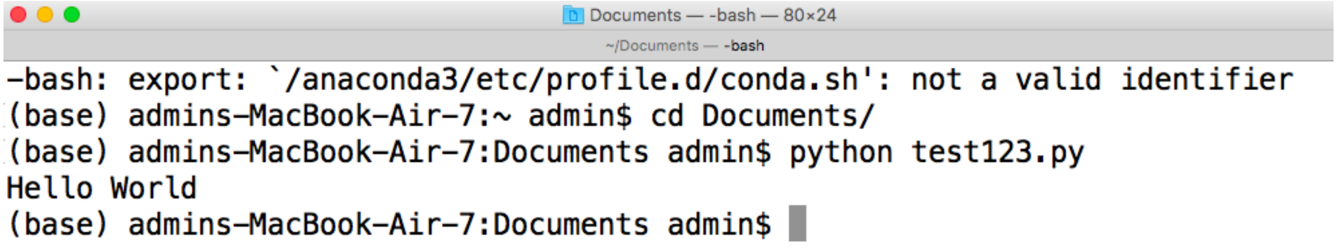
Tags:

Where: Documents

Format: Python files (.py, .pyw)

Figure 1.8 writing and saving code using IDLE editor

you can execute this code by just pressing F5 button in IDLE or using command python test123.py using command line/Terminal. As given below in snapshot-



```
-bash: export: `/anaconda3/etc/profile.d/conda.sh': not a valid identifier
(base) admins-MacBook-Air-7:~ admin$ cd Documents/
(base) admins-MacBook-Air-7:Documents admin$ python test123.py
Hello World
(base) admins-MacBook-Air-7:Documents admin$ 
```

Figure 1.9 Python code execution on the command line

Congratulations!, you done with first python program successfully
These code can be written as an arbitrary.

```
$ python -c ‘print(“Hello World”)’
$ python -c 'print("My name : {0}, city : {1}, phone : {2}".format("Mr ABC","Monrovia", "34453534534"))'
$ python
>>>print(“Hello, ”, end= “\t”)
>>>print(“World”)
>>>print(“Hello, ”, end= “\n”)
>>>print(“Hello, ”, end= “<br>”)
>>>print(“Hello, ”, end= “BREAK”)
>>>import sys
>>>sys.stdout.write(“Hello World”)
```

1.10.1 Basic Operations in python prompt

You can perform basic operation on python prompt such as addition, subtraction, multiplication, division, etc. as given below

```
>>>10*3
30
>>20-15
5
>>>4+5
9
>>>5/2
2.5
>>>5%2
1
>>>5-10
-5
>>>10**3
1000
>>>12//8
1
>>12%8+(12//8)*8
12
>>>10--2
12
>>>a=10
>>>a+=2
>>>a
12
```

1.11 Anatomy of Python program

Python programs are easy to write but it becomes difficult while we develop a big application there we have lots of file and code and it become more difficult to manage while it goes into errors that is why some standards are suggested to avoid this type of difficulties. Some of them rules are

1. How one program is connecting to other programs
2. Code should be written in the isolated form(various small units), so that in the case of can fix it within short period of time.
3. DRY(Do not repeat yourself), never write same code again and again, just read it.
4. OOP (Object Oriented Programming) can make your program more manageable such as classes, methods, objects etc
5. Special configuration and log files must be located at some location for input/output.
6. Always writing comments in the program about its functionality to understand flow of program.

1.12 IDLE

IDLE(Integrated Development and Learning Environment) is an alternative to the command line, as the name suggests its is very useful for developing new code for learning python. This is the default IDE which come with python interpreter but you can user other interpreter too, there are so many advantages to it.

1) Multi-window text editor with syntax highlighting, auto-completion, and smart indent
2) Automatic indentation
3) Integrated debugger with stepping, persisting break poins and call stack visibility
4) Python shell with syntax highlighting

5) Saving the python program as **.py** files and run them and edit them later at any them using IDLE

6) Python program can be run using **F5** key python shell to run interpreter.

1.13 User Input

To get input from user, use the input function(raw_input if you are using python 2.x). This function takes string as an argument. Which displays the prompt and returns a string and wait until user provides input. By default it takes string but it may type caste automatically by input data. And then it may require to convert it into required data type.

```
>>>  name=input(“What is your name?”)
```

1.14 Installing external modules using pip

While it come to installing external modules there are so many way but most efficient is pip. First of all you have to make sure pip is installed in your machine, any way it come with python. Using the following command you can check pip version

```
$ pip –-version          #for python 2
$ pip3 –version          #for python 3
```

On instance with both python 2 and python 3 installed, pip often refers to python-2 and pip3 to python-3. both are different do if you are using python-2 then use pip or python 3 then use pip3. You can do various task using pip some of them are given below

```
$ pip search <query>                                              #to search package
$ pip install [package_name]                                      #to install latest version of package
$ pip install [package_name]=x.x.x                                #to install specific version
$ pip install ‘[package-name]>=x.x.x’                             #to install minimum version package
$ pip –proxy http://<server_address>:<port> install [package_name] #while you are behind proxy server
$ pip list –outdated                                              #list of all outdated packages
$ pip install [package_name] –upgrade                             # to upgrade a package
$ pip install -U pip                                              # to upgrade pip version
$ python -m pip install -U pip
$ py -m pip install -U pip
```

where x.x.x stands for version number of package which one you want to install. There is not utility to upgrade all one time, You need to upgrade packages one after one.

Summary

- Guido Van Rossum has developed python language, it was needed because of the other programming languages were having complex design theory
- There are plenty reason to be popularity of python such as easy to learn and dynamically type caste language.
- Most of the programming languages are considered as difficult because of complex object oriented mechanism but python supports too but you can develop any program without object oriented language. Python internally implements OOP automatically.

- Python is a General purpose language. Market demand about python in various prospective like, salary, number of project developed in most various programming language, current market share of all available programming language
- Python is huge in demand in various sectors that is why we should consider python for mobile application development and what is cross platform mobile applications.
- Python comes with various major release with their major changes yearly, architecture of python application along with its all important component along with their usages.
- Python is available in all the available major operating systems, by keep in mind author has described Step by step Python installation in various available platform like, Windows, Linux, and Mac.
- Beginning of any programming language is major step to write first program, you can create and execute your first python application "Hello World".
- Being a general purpose programming language it can perform Basic arithmetic operations in python scripting. Anatomy of python application
- Any text text editor can be used to write python. Program must be written in UTF-8 encoding. By default python editor which comes with python interpreter as a default IDLE, where you can write python programs.
- Programmers can accept input from user with **input** function. Further these inputs can be used to process information.
- Any programming language can't provide all the features. You can install third party libraries using **pip** command. Pip command is used to manage packages in the development, testing and production environment.

Key terms

Python programs are easy to write but it becomes difficult while we develop a big application then its good design/architecture plays very important role, dependencies compatibility, distributed environment. Downloading and installing a particular python version with respective system requirement. Python installation in various operating system and environment variable settings. Creating and executing hello world program with using IDLE. Various advantage of IDLE with several sample code like to get input from user. In the production environment its require to specific/latest version of libraries, pip is a mostly used tool for module/dependency/library download.

Review Questions

- open browser and go to http://python.org and download python for your operating system and install it. Along with its documentation
- From the python prompt get help from the given terminal use help() command prompt the check all the available documentation.
- Practice basic calculator operation such as addition, subtraction, multiplication, division and modulo on python prompt
- you are driving car at 45 km/h, how long it will take to travel 10 Km distance.

Exercise

Tick the correct option

Q.1 Python supports multiple programming paradigms like procedural, OOP(Object Oriented Programming), functional?

a) Procedural
b) OOP (Object Oriented Programming)
c) Functional Programming
d) All of the above

Answer: d) All of the above

Q.2. Python is very slow because of

a) JVM
b) JIT
c) Compiler
d) All of the above

Answer: d) All of the above

Q.3. Which key is used to run/execute python program?

a) F5
b) F6
c) F7
d) F8

Answer: a) F5

Q.4. Python's official Editor is __________?

a) PyCharm
b) Netbeans
c) IDLE
d) Eclipse

Answer: c) IDLE

Q.5. Python was released in

a) 1990
b) 1991
c) 1989

d) 1992

Answer: b) 1991

Q.6. What is pinning?, pip freeze command can be

a) Adding dependency in Readme file

b) Adding dependency in py file

c) Adding dependency in xml file

d) Adding dependency in requirements.txt

Answer: d) Adding dependency in requirements.txt

Q.7. Which of the following command is used to all the production dependencies in one file?

a) ls , dir

b) pip, freeze

c) cd, mkdir

d) python, pip

Answer: b) pip, freeze

Q.8. DRY stands for?

a) Do not Repeat Yourself

b) Design Reference Year

c) Defensive Rushing Yards

d) Dependent Responding Y axis

Answer: a) Do not Repeat Yourself

Q.9. Who invented Python?

a) Dennis Ritchie

b) Charles Babbage

c) Guido Van Rossum

d) Sundar Pichai

Answer: c) Guido Van Rossum

Q.10. Which command is used to download python packages or libraries?

a) pip

b) git

c) pit

d) jit

Answer: a) pip

Fill in the Blanks

1. Python language is named after a British Comedy group called ___________________.
2. Python is easiest programming language to ___________in __________ of time.
3. Python basically focuses on___________ instead of basic facts of programming.
4. Some features can be import from python 3 using ______________ in python 2.x.
5. ______ is the most efficient way to install external libraries.

Answers

1. Monty Python’s Flying Circus
2. learn quickly, short period
3. business logic
4. __future__
5. pip