Text Book – Learning Python by Mark Lutz

**About Python**

**Python** is a [widely used](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) [high-level programming language](https://en.wikipedia.org/wiki/High-level_programming_language) for [general-purpose programming](https://en.wikipedia.org/wiki/General-purpose_programming_language), created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991.

**History of Python**

The [Python programming language](https://en.wikipedia.org/wiki/Python_programming_language) was conceived in the late 1980s, and its implementation was started in December 1989 by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) at [CWI](https://en.wikipedia.org/wiki/Centrum_Wiskunde_%26_Informatica) (The ***Centrum Wiskunde & Informatica*** a National Research Institute for Mathematics and Computer Science) in [the Netherlands](https://en.wikipedia.org/wiki/The_Netherlands) as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_programming_language) capable of [exception handling](https://en.wikipedia.org/wiki/Exception_handling) and interfacing with the [Amoeba operating system](https://en.wikipedia.org/wiki/Amoeba_distributed_operating_system). 

Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, [*Benevolent Dictator for Life* (BDFL)](https://en.wikipedia.org/wiki/Benevolent_Dictator_For_Life).

Python was named for the BBC TV show [*Monty Python's Flying Circus*](https://en.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus).

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the General Public License (GPL).

Available for download from http://www.python.org

**Definition of Python.**

General purpose programming language that blends procedural, functional and object oriented paradigms.

Python is free, its a open source programming language.

Can be used for writing standalone programs and scripting applications.

**ADVANTAGES OF PYTHON**

**1. Software quality** bcoz - its readable, reusable, maintainable, it has deep support for advanced software reuse mechanism, such as object oriented and function programming.

**2. Developers productivity** - 1/3 SIZE when compared with other programming languages

- No lengthy steps of compilation

- Python code runs immediately

- Less to type, less to debug, less to maintain

**3. Program portability** - Can be run on major computer platforms. Copy paste(wins-Linux)

- you can port operating system interfaces.

- Multiple options for coding GUI's, database access, web based systems

**4. Support Libraries** - Python has a standard library, It is prebuilt with portable functionality.

- Can be extended with homegrown libraries and third party application s/w.

-Ex : NumPy extension - works as matlab.

**5. Component Integration** - Python can be used to invoke C and C++ code.

- It can be integrated with Java and .NET Components.

-Can interact over the n/w with interfaces like SOAP,XML-RPC, CORBA

**FEATURES OF PYTHON**

1. **ITS OBJECT ORIENTED AND FUNCTIONAL**

- Python supports polymorphism, inheritance, operator overloading.

- It's easy to learn OOP concepts in python

- Python supports both object oriented programming and functional programming paradigms.

**- generators,**

a=5 <int object>

a

a= ’5.6’ <Str object>

A = <expr> ---> Sometimes ---> II unit

A --- needed statements to execute a Generator object

**comprehensions**

Ex: Generate an even number in a given range 1- 100: No of lines in C? 10 - Python 1

**Lambdas. Or Anonymous Function**

**General\_Function <func\_name>(parameters):**

**{**

**Statements1…**

**2**

**..**

**}**

**<func\_name>(parameters);**

**a=lambda <parameters>:<expression>**

**a(parameter)**

1. **ITS FREE** - you can fetch the entire pythons source code. You can also sell it
2. **ITS PORTABLE** - Standard implementation of python is written in portable ANSI C and it runs on every major platform currently in use.
3. **ITS POWERFUL**

- **Dynamic typing** - It doesn't require complicated size and type declarations.

- **Automatic memory management** - Python automatically allocates objects and reclaims when they are no longer used. Python keeps track of low level memory details.

- **Programming-in-the-large support - for building larger system.** Python includes tools such as modules, classes exceptions. Use OOP concept to reuse and customize code.

- **Built in object types** - Provides commonly used data structures such as lists, dictionaries and strings.(tuples and sets)

- **Built in tools** - It has powerful standard operations such as slicing, concatenation, sorting, mapping and more

- **Library utilities** - large code of precoded library tools that support from regular expression (re) matching to networking.

- Third-party utilities - Developers contribute precoded tools that supports tasks beyond those supported by its built-ins.

1. **ITS MIXABLE** - Pythons programs can easily be "**glued**" to components written in other languages in a variety of ways. Example : **Pythons C API (Application Program Interface)** lets C programs call and called by python programs flexibly.
2. **ITS RELATIVELY EASY TO USE** - To run a python program, you simply type it and run it.

Compared to C, C++ JAVA. There a re no intermediate compile and run steps.

- Provides simple syntax and powerful built in tools.

1. **ITS RELATIVELY EASY TO LEARN**

Its named after Monty python - Cartoon 1970s BBC comedy series Monty pythons flying circus,

monty python and the holy grail.

**IS PYTHON A SCRIPTING LANGUAGE?**

Is a general purpose programming language, often applied in scripting roles.

Shell tool : used to launch other programs, file processing, user creation, server launching, etc.

Control language : Control other application components. - Low level access.

No need of recompilation.

No need of having knowledge about integrated components.

Ease of Use: Because of its simple feature set.

**DOWNSIDE(Disadvantages) OF PYTHON**

Why might you not want to use Python in an application?

Execution speed.

Compilation

source code --> byte code

Python is not compiled to binary machine code

**APPLICATIONS OF PYTHON**

What can you do with Python?

**1. Systems programming** - you can write portable, maintainable system administration tools.

- Some of the python tools can search files, directories, launch other programs.

- Python standard library comes with POSIX bindings which makes support easier for all the OS Tools.

**2. GUI** - Python has a standard object oriented interface to TkGUI API called Tkinter.

- Tkinter is used to implement portable GUIs with native look and feel.

- A free extension package PMW adds advanced widets to tkinter toolkit.

- wxPython GUI API based on C++ library an alternative tool for constructing portable GUIs

- Dabo toolkit for GUI

- QT - PyQT

- GTK -PyGTK

- MFC - PyWIN32

-.NET - Iron Python

-Swings - Jython or JPype

**3. Internet scripting** - Python comes with standard internet modules.

- Python can perform a variety of networking tasks.

- Extracting information sent to server-side CGI scripts

- Transfer files by FTP

- Parse and generate XML and JSON documents.

- Send receive compose and parse email

- fetch web pages by URL

- to generate HTML files we can use HTMLGen

- mod\_python runs python efficiently on Apache web server

- Jython - used for coding server side applets.

- Django, TurboGears, web2py, pylons, Zope, WebWare support for quick construction of full-featured and production quality websites with python.

- Silverlight tool of microsoft.

**4. Component Integration** - Integrating a C library into python enables python to test and launch

the library's components, and embedding python in a product enables onsite customizations to be coded without having to recompile the entire product. SWIG and SIP tools use cython for scripts.

**5. Database Programming** - There is a python interface for all commonly used relational databases -

Sybase, Oracle, informix, ODBC, MySQL, PostgreSQL,SQLite.

portable database API - used to access SQL Database systems

Pickle module - non-sql databases can be accessed, it allows programs to easily save and restore entire python objects to files and file like objects.

Third Party systems - ZODB, Durus provide OODB for python scripts

PyMango an interface to MangoDB

Pythons Json module to create Json style document database

**6. Rapid prototyping**. - We can initially prototype systems in python then move selected components to a compiled language such as C or C++. As Python and C look the same.

**7. NUMERIC AND SCIENTIFIC PROGRAMMING** - NumPy - Matlab , is a numeric programming tool that can often replace existing code written in C C++.

Numeric tools for python support animation, 3D visualization, parallel processing

SciPy - Provides library for scientific programming.

**8. Gaming, images ,data mining, Robotics. - (tools )**

Game programming and multimedia with pygame, cgkit, pyglet, PySoy, Panda3D, and others

Serial port communication on Windows, Linux, and more with the PySerial extension

Image processing with PIL and its newer Pillow fork, PyOpenGL, Blender, Maya,and more

**What is the Python interpreter?**

* Python is a software package also called as interpreter.
* Interpreter is a kind of program that executes other programs.
* An interpreter reads a high-level program and executes it
* Interpreter is a layer of software logic between your code and the computer hardware on your machine.
* Interpreter is implemented in C, a Set of Java Classes or something else.
* When you install python it generates many components - minimally, an interpreter and a support library.
* Interpreter takes the form of executable program.

**How Python Runs Programs (Interpreters view)**

**1. BYTECODE COMPILATION.**

- python compiles and translates each of your source statements into a group of byte code instructions by decomposing them into individual steps.

- Byte codes are stored in files with .pyc extension.

In 3.2 and later versions It is stored in a subdirectory name \_\_pycache\_\_

- Next time if you run the program, if source code is not changed, then .pyc is loaded and program is executed fastly.

- source changes - python checks last modified time stamps

- Python versions - imports also check with which version it was compiled.

- If byte code cannot be written to HDD then it will be saved in memory and executed later the code will be discarded.

-Note - Byte code is only saved for the files that are imported.

**2. Python Virtual Machine**

- PVM is a big code loop that iterated through your byte code instructions one by one, to carry out their operations.

- PVM is a runtime engine of python

- It is component that runs your scripts

- Also called as python interpreter.

**3. PERFORMANCE IMPLICATIONS.**

-Python byte code is not a binary m/c code, byte code is something which is python specific

representation.

- This is why python may not run as fast as C C++

- byte code instructions require more work than CPU instructions.

- PVM loop takes more amount of time.

- Python code runs at speeds somewhere between those of a traditional compiled language and a

traditional interpreted language

**4. DEVELOPMENT IMPLICATIONS**

-In Python execution model there is really no distinction between the development and execution

environments.

- the systems that compile and execute your source code are really one and the same.

- There is no need to precompile and link before execution may begin; simply type and run the code.

-The eval and exec built-ins, for instance, accept and run strings containing Python program code.

-This structure is also why Python lends itself to product customization— because Python code can be changed on the fly, users can modify the Python parts of a system onsite without needing to have or compile the entire system’s code.

**EXECUTION MODEL VARIANTS**

Prominent implementations available today

* Cpython : The original and standard, implementation of python is usually called Cpython/Python. Coded in portable ANSI C language code.
* Jython : Python for Java
* Jython consists of Java classes that compile Python source code to Java byte code and then route the resulting byte code to the Java Virtual Machine (JVM).
* IronPython : Python for .NET
* Stackless: Python for concurrency provides efficient multiprocessing options. The microthreads that Stackless adds to Python are an efficient and lightweight alternative to Python’s standard multitasking tools such as threads and processes, and promise better program structure, more readable code, and increase programmer productivity
* PyPy: Python for speed

**EXECUTION OPTIMIZATION TOOLS**

Cython:

– A Python/C hybrid :Cython can be useful both for wrapping external C libraries and for coding efficient C extensions for Python

Shed Skin: A Python-to-C++ translator.

– it attempts to translate Python source code to C++ code, which your computer’s

– C++ compiler then compiles to machine code.

Psyco: The original just-in-time compiler

– Psyco is an enhancement to the PVM

– Psyco program runs to translate portions of the program’s bytecode all the way down to true binary machine code for faster execution.

FROZEN Binaries :

– Turn your Python programs into true executables, known as frozen binaries in the Python world.

**Who Uses Python Today?**

Python is also being applied in real revenue generating products by real companies.

* Google makes extensive use of Python in its web search systems.
* The popular YouTube video sharing service is largely written in Python.
* The Dropbox storage service codes both its server and desktop client software primarily in Python.
* The Raspberry Pi single-board computer promotes Python as its educational language.
* EVE Online, a massively multiplayer online game (MMOG) by CCP Games, uses Python broadly.
* The widespread BitTorrent peer-to-peer file sharing system began its life as a Python program.
* Industrial Light & Magic, Pixar, and others use Python in the production of animated movies.
* ESRI uses Python as an end-user customization tool for its popular GIS mapping products.
* Google’s App Engine web development framework uses Python as an application language.
* The IronPort email server product uses more than 1 million lines of Python code to do its job.
* Maya, a powerful integrated 3D modeling and animation system, provides a Python scripting API.
* The NSA uses Python for cryptography and intelligence analysis.
* iRobot uses Python to develop commercial and military robotic devices.
* The Civilization IV game’s customizable scripted events are written entirely in Python.
* The One Laptop Per Child (OLPC) project built its user interface and activity model in Python.
* Netflix and Yelp have both documented the role of Python in their software infrastructures.
* Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm, and IBM use Python for hardware testing.
* JPMorgan Chase, UBS, Getco, and Citadel apply Python to financial market forecasting.
* NASA, Los Alamos, Fermilab, JPL, and others use Python for scientific programming tasks.

**IDEs**

* IDLE
* Eclipse and PyDev
* Komodo
* NetBeans IDE for Python
* PythonWin
* Wing, Visual Studio, and others
* Wing IDE, Microsoft Visual Studio via a plug-in, and PyCharm, PyScripter, Pyshield, and Spyder