MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

LAB-REPORT

Report no : 03

Report Name: Introduction to Python Lab

Course code : ICT-3208

Course title : Network Planning and Designing Lab

Date of Submission : 21-August-2020

Submitted By

Name: Md. Nayan Ali
ID: IT-16062
3rd year 2nd semester
Session: 2015-2016
Dept. of ICT
MBSTU.

Submitted To

Mr. Nazrul Islam
Assistant Professor,
Dept. of ICT, MBSTU.

Objectives:

i) Setup python environment for programing on Ubuntu.
ii) Learn the basics of python.
iii) Create and run basic examples using python.

Introduction to Python Lab

1. What is Python? Give the definition of Python.

Answer: Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

2. What are the main features of Python? Describe them briefly.

Answer: The main features of Python are given below:

 Simple: Python is a simple and minimalistic language. This pseudo-code nature of Python is one of its greatest strengths.
 Easy to Learn: Python is extremely easy to get started with. Python has an extraordinarily simple syntax.
 Free and Open Source: Python is an example of FLOSS (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read it's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge.
 High-level Language: When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.
 Portable: Due to its open-source nature, Python has been ported (i.e. changed to make it work on) to many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.
 Multi-Platform: Python can be used on Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and even PocketPC.
 Interpreted: Python does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called byte codes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!
 Object Oriented: Python supports procedure-oriented programming as well as object oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality.

 Extensible: If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your programing C or C++ and then use them from your Python program.
 Embeddable: You can embed Python within your C/C++ programs to give 'scripting' capabilities for your program's users.
 Extensive Libraries: The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), Tk, and other system-dependent stuff. Remember, all this is always available wherever Python is installed.
 Other Libraries: Besides, the standard library, there are various other high-quality libraries such as:
o wxPython [http://www.wxpython.org],
o Twisted [http://www.twistedmatrix.com/products/twisted],
o Python Imaging Library [http://www.pythonware.com/products/pil/index.htm]

Setup of Python environment on Ubuntu:

Step-1(Update): Logged into your Ubuntu 18.04 server as a sudo non-root user, first update and upgrade your system to ensure that your shipped version of Python 3 is up-to-date.

```
ruhan@ruhan-VirtualBox:~$ sudo apt update
[sudo] password for ruhan:
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu bionic InRelease
Hit:3 http://bd.archive.ubuntu.com/ubuntu bionic InRelease
Get:4 http://bd.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metada
ta [49.0 kB]
Get:6 http://bd.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Me
tadata [56.0 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11
Metadata [2,464 B]
Get:9 http://bd.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Metad
ata [295 kB]
Get:10 http://bd.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 DEP-11
Metadata [285 kB]
Get:11 http://bd.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 DEP-1
1 Metadata [2,468 B]
Get:12 http://bd.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-1
1 Metadata [9,292 B]
Fetched 951 kB in 7s (146 kB/s)
```

Step-2(Install supporting software): The software-properties-common package gives you better control over your package manager by letting you add PPA (Personal Package Archive) repositories. Install the supporting software with the command:

```
ruhan@ruhan-VirtualBox:~$ sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties software-properties-gtk
The following packages will be upgraded:
  python3-software-properties software-properties-common
  software-properties-gtk
3 upgraded, 0 newly installed, 0 to remove and 356 not upgraded.
Need to get 0 B/96.7 kB of archives.
After this operation, 3,072 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Step-3(Add deadsnakes ppa to the source list): Deadsnakes is a PPA with newer releases than the default Ubuntu repositories. Add the PPA by entering the following:

```
Get:1 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease [15.9 kB]
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://bd.archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu bionic InRelease
Hit:5 http://bd.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:6 http://bd.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:7 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic/main amd64 Packages [29.5 kB]
Get:8 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic/main i386 Packages [29.6 kB]
Get:9 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic/main Translation-en [6,904 B]
Fetched 81.9 kB in 3s (25.7 kB/s)
Reading package lists... Done
ruhan@ruhan-VirtualBox:~$
```

Step-4(Install Python3): Now you can start the installation of Python 3.8 with the command:

```
ruhan@ruhan-VirtualBox:~$ sudo apt install python3.8
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libpython3.8-minimal libpython3.8-stdlib python3.8-minimal
Suggested packages:
  python3.8-venv python3.8-doc binfmt-support
The following NEW packages will be installed:
  libpython3.8-minimal libpython3.8-stdlib python3.8 python3.8-minimal
0 upgraded, 4 newly installed, 0 to remove and 356 not upgraded.
Need to get 4,765 kB of archives.
After this operation, 18.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Step-5(Installed successfully): Allow the process to complete and verify the Python version was installed successfully.

```
ruhan@ruhan-VirtualBox:~$ python3 --version
Python 3.6.8
ruhan@ruhan-VirtualBox:~$
```

Conclusion: From this lab I've learned that the installing the python process in Ubuntu. Also learn the basic installing command of python in Ubuntu.