Installation of Python on Linux

Using the Package Manager

1. If you are using Ubuntu Linux, at the command prompt type:

```
$ sudo apt-get install python3
```

2. If you are using Fedora, at the command prompt type:

```
$ sudo yum install python3
```

3. The most recent version of Python 3 will be downloaded and installed. To verify the installation, type:

```
$ Python
```

The Python prompt (>>>) will appear.

From source

You can install Python from its source code if you want to really customize the binary by setting certain options or flags during the build process.

1. Download the source from https://www.python.org/ftp/python/3.6.2/Python-3.6.2rc1.tgz
2. Type the following commands to extract and install Python from its source:

```
$sudo tar xzf python-3.6.2.tar.xz

$./configure

$sudo make install
```

3. To verify the installation, in the terminal type:

```
$ Python3
```

4. The Python prompt (>>>) will appear.

Installation of Python on Mac OS X

Using a wizard-based installer

1. Go to the Python website, https://www.python.org/ftp/python/3.6.2/python-3.6.2-macosx10.6.pkg and download the required version.
2. Run the downloaded file and follow the instructions in the installation wizard.

Using Homebrew

Homebrew is a package manager that lets you install, update, and uninstall packages from the command line on the Mac OS.

1. Homebrew depends on Apple's Xcode package, so run the following command to install Xcode first:

```
$ xcode-select --install
```

2. Next, install Homebrew by following the instructions on their website, https://brew.sh/

3. After installing Homebrew, from the prompt in the terminal type the following command to install Python.

```
$ brew install python3
```

4. To verify the installation, type:

```
$ Python3
```

The Python prompt (>>>) will appear.

Installation of Python on other platforms

You can install Python for other platforms such as MS-DOS, OS2 or Solaris from the Python website, https://www.python.org/download/other/

Alternative Python implementations

The standard distribution of Python (available on python.org) is developed in C Language and is often called CPython. This recommended standard is tweaked to optimize its performance for specific applications. These implementations combine ease of development of Python and rich libraries of other platforms such as .net or Java. Some such alternative implementations are:

- IronPython (Python running on .NET framework. Developed in C#)
- Jython (Developed in Java. Python running on the Java Virtual Machine. Capable of using rich Java library in Python program)
- PyPy (A fast python implementation with a JIT compiler)
- Stackless Python (Branch of CPython supporting microthreads)
- MicroPython (Python running on micro controllers)

Commercial distributions of Python

While community versions are open source and intended to use on as is basis, commercial versions of Python ensure guarantee of service and maintenance of the products. Also, when modules and packages are installed from repositories, it may lead to version conflict in a community version. However, commercial versions ensure that the distribution is self-contained and without compatibility issues. Some of the commercial distributions of Python include:

- ActiveState ActivePython (commercial and community versions, including scientific computing modules)
- pythonxy (Scientific-oriented Python Distribution based on Qt and Spyder)
- winpython (WinPython is a portable scientific Python distribution for Windows)
- Enthought Canopy (a commercial distribution for scientific computing)
- PyIMSL Studio (a commercial distribution for numerical analysis)
- Anaconda Python (a full Python distribution for data management, analysis and visualization of large data sets)