

How to Install Latest Python on Ubuntu 18.04

Updated March 16, 2021 by Hayk Gevorgyan



In this tutorial, we are going to show how to install the latest python on Ubuntu 18.04 machine using the terminal. Also, we will create a virtual environment and a simple program and run it. At the moment of writing this article, the latest stable version of python was 3.6.5 but the process of installing allows you to install the latest version by changing just the version number.

Note that Ubuntu 18.04 has python 3 preinstalled. However, we are going to show how to install it, update and uninstall.

How to install latest python

Before installing the latest version of python check the version of python currently installed on your machine. To do so you can run:

```
python3 -V
```

or

```
python3 --version
```

If there are several versions installed on your machine you can run the following command to view all versions:

```
apt list --installed | grep python
```

If you want to install the latest version of python3 on your Ubuntu18 machine you can run:

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

or if it's already preinstalled, you can run the command below to update to latest version.

```
sudo apt-get upgrade python3
```

However, if you want to install it manually you can use the source code. To do so, follow the instruction below.

How to install python from source

To install the latest python from source, get the download link of the latest version from [python's official website](#) and download it to your Ubuntu machine. To do so you can run:

```
wget https://www.python.org/ftp/python/3.6.5/Python-3.
```

Once it is downloaded, extract the archive using the command below:

```
tar -xvf Python-3.6.5.tgz
```

Change your directory to the newly created Python3.6.5 folder by running:

```
cd Python-3.6.5
```

and run the configure script to check the build

```
./configure
```

Once the check has been completed run the following commands in the mentioned order to install python 3.6.5

```
sudo make
```

```
sudo make install
```

However, if there wasn't need to use `zlib1g-dev` package before, you'll get an error of it's absence, which will look like this:

```
zipimport.ZipImportError: can't decompress data; zlib
Makefile:1099: recipe for target 'install' failed
make: *** [install] Error 1
```

To install the missing package run:

```
sudo apt install zlib1g-dev
```

After package installation run the `sudo make` and `sudo make install` commands once more. After installation is completed check the installed version with the following command:

```
python3.6 -V
```

Output must look like this:

```
Python 3.6.5rc1
```

How to update python to latest version

To update python to the latest version, you can run:

```
sudo apt-get upgrade python3
```

or if you have installed manually from source, you must uninstall the previous version and install the latest version or the one you want. The steps of how to do so are described in the section below.

How to uninstall python

To uninstall python from your Ubuntu 18 machine, run:

```
sudo apt remove python3.6
```

Note that you cannot mention major python3 version as it is being used by the system, but you can uninstall minor version which is preinstalled on your machine.

If you have installed python from source and need to update it, you must uninstall the previous one and repeat the installation steps which are described above in this article. As this method if installation locates python files in `/usr/local/bin` path, you must remove python files from that path. To do so you can run the following commands:

```
sudo rm /usr/local/bin/py*
sudo rm /usr/local/bin/pip*
sudo rm /usr/local/bin/idle*
sudo rm /usr/local/bin/2to3*
sudo rm /usr/local/bin/easy_install-3.6
```

Once all files are removed you can proceed to the installation of newer version.

How to setup virtual environment for python3

Virtual environment for python is an isolated space on your computer for Python projects, which allows each of your projects to have its own set of dependencies and don't disturb any other project. Inside the virtual environment, you can use `python` and `pip` commands instead of `python3` and `pip3` accordingly. Outside of an environment, you must use `python3` and `pip3` commands for running Python 3.

Following the steps below you can setup a virtual environment for python3 and use it to develop applications.

First of all you need to check if `pip` (python package manager) installed along with your python installation. If you installed python from source, `pip` is being installed along with python. To check `pip` version run:

```
pip3 -V
```

Output must look like the one below:

```
pip 9.0.1 from /usr/lib/python3/dist-packages (python
```

If `pip` isn't installed on your machine, run the following command to install it:

```
sudo apt install python3-pip
```

Once `pip` is installed you can run the following command to install python packages:

```
pip3 install [package-name]
```

Besides `pip` you need some more packages that need to be installed as prerequisites. To install them all, run:

```
sudo apt-get install build-essential libssl-dev libffi
```

Once all prerequisites are installed you can proceed to virtual environment creation. To do so follow the steps below:

Install `python3` virtual environment package by running the command below:

```
sudo apt install -y python3-venv
```

Create folder for environments and change your working directory to it. To do so, run:

```
mkdir environments && cd environments
```

Create a virtual environment and give it a name which will best describe the environment purpose:

```
python3 -m venv my_first_env
```

Actually, the command above create a directory with needed files. To view them type:

```
ls my_first_env
```

The output must look like the one below:

```
bin include lib lib64 pyvenv.cfg share
```

To be able to use the newly created environment you must run:

```
source my_first_env/bin/activate
```

After activating the environment you must see the environment name in brackets in front of your prefix. E.g.

```
(my_first_env) root@ubuntu:~/environments#
```

To leave the environment just type the command below:

```
deactivate
```

How to create simple python application within the virtual environment

After virtual environment is created we can create a simple application inside it and test if it works. Our application will output "I am new to Python!" every time we run it. First of all we need to activate the environment or, if you want, enter inside the virtual environment. To do so, change your working directory to environments folder and run the command to activate the desired environment:

```
cd ~/environments
```

```
source my_first_env/bin/activate
```

Create a file with .py extension via your favorite text editor and give it a name:

```
vim my_first_app.py
```

Add the following text in the file and save it:

```
print("I am new to Python!")
```

Now every time you run your application with this command:


```
python my_first_app.py
```

you'll see the output:

```
I am new to Python!
```

Read Also:

- [How to Install Golang 1.8 on Debian 9 Stretch](#)

Python 3 comes already installed with Ubuntu 18.04 and it's very easy to update it to latest once there is a need. However, if you are using Ubuntu 18's minimal version you must install python manually and there are two ways to install it. One is using apt, which will install minimal packages of python3 and the other using source code which will install the full version of python 3.