

Python

- [Default Python versie wijzigen in Debian 10](#)
- [Python 3.8 installeren in Debian 10 \(Buster\)](#)
- [Python installeren vanuit source in Debian 10](#)
- [Virtualenv gebruiken in Python3](#)

Default Python versie wijzigen in Debian 10

Je kunt meerdere Python versies installeren in Linux.

Om te zien welke versie op dit moment actief (default) is kan je het volgende uitvoeren:

```
python -V
```

Om te zien welke versies geïnstalleerd zijn in Linux kan je het volgende commando uitvoeren:

```
ls /usr/bin/python*
```

Prioriteren van de verschillende Python versies

Wanneer je met bovenstaand commando weet welke Python versies zijn geïnstalleerd, dan kan je met onderstaand commando de prioriteit aangeven:

```
update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1  
update-alternatives --install /usr/bin/python python /usr/bin/python3.8 2
```

In bovenstaand geval heeft Python versie 3.8 de hoogste prioriteit, dus dan is dit de default Python versie.

Wanneer je een Python versie hebt [geïnstalleerd vanuit source](#) (zelf gecompileerd), dan kan je deze versie toevoegen met:

```
update-alternatives --install /usr/bin/python python /usr/local/bin/python3.9 3
```

Hiermee wordt Python3.9 op prioriteit 3 gezet en heeft dus in dit geval de hoogste prioriteit. Deze versie is dan de actieve Python versie.

Switchen tussen default Python versie

Wanneer je een andere Python versie de default versie wilt maken, voer dan onderstaand commando uit:

```
update-alternatives --config python
```

Je kunt de default versie weer controleren met: `python -V`

Python 3.8 installeren in Debian 10 (Buster)

Python 3.8 installeren

In Debian 10 hebben we standaard alleen de beschikking over Python 2.7 en Python 3.7. Je kan met onderstaand commando controleren welke versies van Python geïnstalleerd zijn in Debian:

```
ls /usr/bin/python*
```

De actieve Python versie kan je controleren met:

```
python -V
```

Nu kunnen we Python vanuit de source installeren, maar we hebben ook de beschikking over een repository die beschikbaar is gesteld door Pascal Roeleven. Het voordeel is dat we dan op de normale manier de Python 3.8 pakketten kunnen installeren.

Voordat we hiermee verder gaan moeten we eerst nog een paar benodigde pakketten installeren:

```
apt-get install gpg wget
```

Daarna kunnen we de GPG key installeren die nodig is:

```
wget https://pascalroeleven.nl/deb-pascalroeleven.gpg  
apt-key add deb-pascalroeleven.gpg
```

Nu moeten we nog de sources.list aanmaken. Hiervoor maken we een nieuw bestand aan met de tekst editor:

```
nano /etc/apt/sources.list.d/python3.8-pascalroeleven.list
```

Hier plaatsen we het volgende in:

```
deb http://deb.pascalroeleven.nl/python3.8 buster-backports main
```

Hierna een update uitvoeren:

```
apt-get update
```

En dan uiteindelijk Python 3.8 installeren:

```
apt-get install python3.8 python3.8-venv python3.8-dev
```

BRON:

[Python 3.8 backport for Debian buster](#)

Python installeren vanuit source in Debian 10

De nieuwste Python versie is vrij eenvoudig te installeren vanuit de broncode (source). In onderstaande stappen wordt beschreven hoe men dit kan doen.

Benodigde pakketten installeren

```
apt-get install build-essential libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev libffi-dev zlib1g-dev
```

Downloaden Python broncode

Op het moment van schrijven is Python 3.9.4 de nieuwste versie. Deze gaan we downloaden en uitpakken.

```
cd /usr/src
wget https://www.python.org/ftp/python/3.9.4/Python-3.9.4.tgz
tar xzf Python-3.9.4.tgz
```

Python compileren en installeren

```
cd Python-3.9.4
./configure --enable-optimizations
make altinstall
```

Het laatste commando “**make altinstall**” zorgt er voor dat de nieuwe versie naast de bestaande Python versies wordt geïnstalleerd. Doet men dit niet, dan wordt de bestaande versie overschreven.

Python 3.9.4 is nu geïnstalleerd in /usr/local/bin.

Controleren Python versie

Je kunt nu de Python versie controleren met:

```
python3.9 -V
```

Python verwijderen

Wanneer Python op deze manier wordt geïnstalleerd, dan kan men niet met een commando de installatie weer ongedaan maken. Hiervoor dient men de bestanden en directories met de hand te verwijderen. Onderstaande commando's zorgen ervoor dat alles verwijderd wordt:

```
rm -f /usr/local/bin/2to3-3.9
rm -f /usr/local/bin/easy_install-3.9
rm -f /usr/local/bin/idle3.9
rm -f /usr/local/bin/pip3.9
rm -f /usr/local/bin/pydoc3.9
rm -f /usr/local/bin/python3.9
rm -f /usr/local/bin/python3.9-config
rm -rf /usr/local/bin/include/python3.9
rm -f /usr/local/lib/libpython3.9.a
rm -rf /usr/local/lib/python3.9
rm -rf /usr/local/lib/pkgconfig
rm -f /usr/local/share/man/man1/python3.9.1
```

BRON:

[How to Install Python 3.8 on Ubuntu, Debian and LinuxMint](#)

Virtualenv gebruiken in Python3

Virtualenv is een tool om een geïsoleerde Python omgeving te maken.

Deze omgeving heeft zijn eigen installatie directory die geen libraries deelt met andere virtuele omgevingen.

Verschil tussen virtualenv en venv

venv is een Python package dat standaard aanwezig is in Python3 (niet in Python2).

virtualenv is een python library die meer functionaliteiten biedt dan **venv**.

Via onderstaande link kan je zien wat de verschillen zijn tussen deze twee omgevingen:

- <https://virtualenv.pypa.io/en/stable>

Virtualenv installeren met behulp van pip3

We gaan er van uit dat Python3 is geïnstalleerd vanuit de source zoals beschreven in [dit artikel](#).

Op het moment van schrijven is dit Python3.9.4. Je kan met onderstaand commando controleren welke versie Python actief is:

```
python3.9 -V
```

Met onderstaand commando kan je zien wat het pad is waar Python3.9 is geïnstalleerd:

```
which python3.9
```

In ons geval is dat /usr/local/bin/python3.9

We gaan nu eerst de package **pip** updaten. **pip** is de Python package installer.

```
python3.9 -m pip install --upgrade pip
```

Hierna kunnen we de **virtualenv** package gaan installeren:

```
pip3 install virtualenv
```

Virtuele Python omgeving maken

We kunnen nu een virtuele omgeving aanmaken met behulp van het commando **virtualenv**. Let er op dat je ingelogd bent als de gebruiker waarvoor je de virtuele omgeving wilt maken. In onderstaand voorbeeld log ik in vanuit de root console als de gebruiker alex:

```
sudo -su alex
```

En maak dan nu de virtuele omgeving aan met:

```
virtualenv -p /usr/local/bin/python3.9 /home/alex/pythonvenv
```

Je krijgt dan een vergelijkbare output te zien zoals hier onder:

```
alex@tux:~$ virtualenv -p /usr/local/bin/python3.9 /home/alex/pythonvenv

created virtual environment CPython3.9.4.final.0-64 in 274ms
creator CPython3Posix(dest=/mnt/sdb3/home/alex/pythonvenv, clear=False, no_vcs_ignore=False,
global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy,
app_data_dir=/mnt/sdb3/home/alex/.local/share/virtualenv)
added seed packages: pip==21.0.1, setuptools==56.0.0, wheel==0.36.2
activators
BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
```

Je kan de virtuele omgeving een willekeurige naam geven. In bovenstaand voorbeeld is dit dus pythonvenv.

Er wordt dan automatisch een directory aangemaakt met deze naam waarin de virtuele omgeving in wordt gemaakt.

Wanneer je daarin kijkt zie je de volgende directories staan:

```
alex@tux:~/pythonvenv$ ls -l
total 12
drwxr-xr-x 2 alex alex 4096 Apr 23 11:59 bin
drwxr-xr-x 3 alex alex 4096 Apr 23 11:59 lib
-rw-r--r-- 1 alex alex 227 Apr 23 11:59 pyenv.cfg
```

Je ziet dat er binnen de virtuele omgeving de directories **bin** en **lib** zijn aangemaakt, evenals een configuratie bestand **pyenv.cfg**.

De virtuele omgeving heeft nu zijn eigen python en pip versie en de benodigde libraries.

Het activeren van de virtuele python omgeving kan je doen met:

```
source /home/alex/pythonvenv/bin/activate
```

Je ziet dan ook de prompt veranderen in:

```
(pythonvenv) alex@tux:~$
```

De prefix tussen haakjes (pythonvenv) geeft aan dat je in de virtuele python omgeving werkt. Het de-activeren kan nu met het commando:

```
deactivate
```

Je krijgt dan weer de normale prompt terug.

Verwijderen van de virtuele Python omgeving

Het verwijderen van de virtuele Python omgeving is niets meer dan het verwijderen van de complete directory inclusief onderliggende directories en bestanden:

```
rm -rf /home/alex/pythonvenv
```

BRON:

[Installing and using virtualenv with Python 3](#)