PROGRAMMING

How To Install the Latest Python Version on Raspberry Pi?

Raspberry Pi and Python work well together, and Python comes pre-installed on your Raspberry Pi OS. But as often with computers and programming, it's not always that simple. In this article, I'll tell and show you everything you need to know about the Python versions on your Raspberry Pi.

The only way to install the latest Python version on Raspberry Pi OS is to download it from the official website and install it from sources. Raspberry Pi OS repositories are generally late from a few versions.

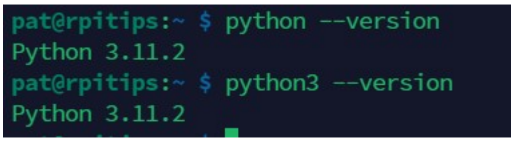
As always, I'm doing this on my Raspberry Pi, so you won't have to face bugs and errors yourself. Follow my recommendations below and everything should work on the first try!

Table of Contents

By the way, if you get overwhelmed as soon as Python is required for a project, I recommend checking out my e-book "Master Python on Raspberry Pi". It will guide you step-by-step to learn the essential concepts (and only the essential concepts) required to achieve any project in the future. Raspberry Pi without Python is like a car without an engine, you miss all the fun parts. **Get 10% off by downloading it today!**

How to know which Python version is installed

To find the exact version number, use the command line "python –version" or "python3 –version".



```
pat@rpitips:~ $ python --version
Python 3.11.2
pat@rpitips:~ $ python3 --version
Python 3.11.2
```

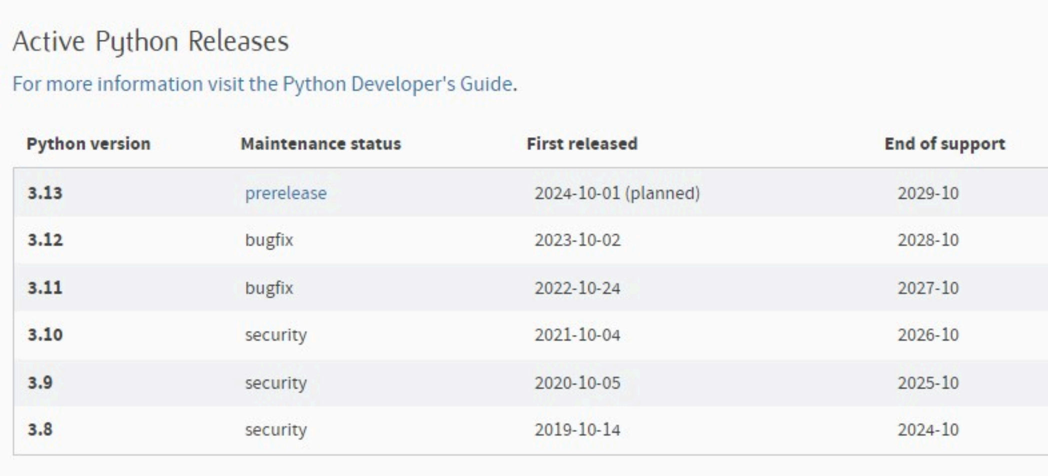
> **Warning**: these commands may not return the same value if you have multiple versions installed (as was the case with Raspberry Pi OS Bullseye and earlier).
> It can be a little disturbing, but yes, there were two versions already installed on your Pi. When you used the command "python" to run a script, you were running it with Python 2. And the "python3" command did the same thing with version 3.

The exact version depends on the latest one available in the Raspberry Pi OS repository. In most cases, with Debian based distribution, these versions are a bit dated. At the time of writing, Python 3.11.2 is two years old, and it's the one pre-installed on Raspberry Pi OS.

By the way, if it is unclear to you why Python is used on Raspberry Pi, you should click on that link to know everything about it.

Find the latest Python version available

The easiest way to find the latest Python release available is to go to the official Python website. On the download page, the latest versions are listed with their release date and maintenance status.



Active Python Releases

For more information visit the Python Developer's Guide.

Python version	Maintenance status	First released	End of support
3.13	prerelease	2024-10-01 (planned)	2029-10
3.12	bugfix	2023-10-02	2028-10
3.11	bugfix	2022-10-24	2027-10
3.10	security	2021-10-04	2026-10
3.9	security	2020-10-05	2025-10
3.8	security	2019-10-14	2024-10

This first table gives you an overview of the latest Pythons versions. As you can see on the Download page, Python 3.11 was released in 2022. In the next section, we'll learn how to update it on your Raspberry Pi.

Install the latest Python version on Raspberry Pi

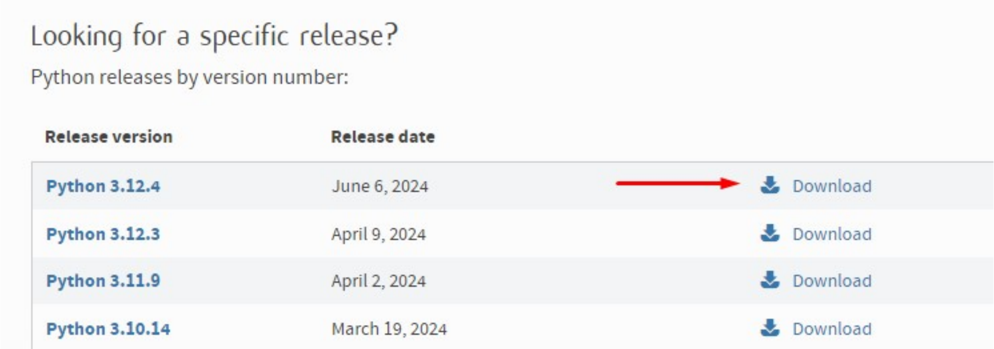
As Raspberry Pi OS is always a few Python versions late, the only way to install the latest Python version on your Raspberry Pi is to download the source code from the official website and install it manually:

- **Step 1: Download the latest version of Python from the official website.**
- **Step 2: Extract the files on the Raspberry Pi.**
- **Step 3: Configure the system to use the latest Python version.**

Let's see how to do this.

Download and extract the latest Python version

- Go to the Python download page.
- Look for the second table on that page "Looking for a specific release?":

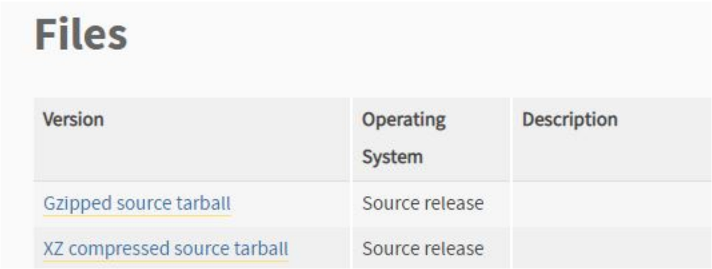


Looking for a specific release?

Python releases by version number:

Release version	Release date	
Python 3.12.4	June 6, 2024	Download
Python 3.12.3	April 9, 2024	Download
Python 3.11.9	April 2, 2024	Download
Python 3.10.14	March 19, 2024	Download

- Click on the "Download" link corresponding to the version you want to install.
  In my case, I will install Python 3.12.4.
- Scroll to the bottom of the next page, and find the list of download links:



Files

Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	

- Right click on "Gzipped source tarball" and choose "Copy link address" from the browser contextual menu.

For the following, you need to open a terminal on Raspberry Pi OS, or connect via SSH to type a few commands. If you need help with the SSH part, you can read my tutorial here which has all the information you might need.

- **Download the latest Python file with:**

  ```
  wget https://www.python.org/ftp/python/3.12.4/Python-3.12.4.tgz
  ```

  Replace the URL with the link you paste in the previous step.

  ```
  wget <url>
  ```

  Related: How to Download Files on Linux Using Terminal
- **Extract the files with:**

  ```
  tar zxvf Python-3.12.4.tgz
  ```

```
tar zxvf <filename>
```

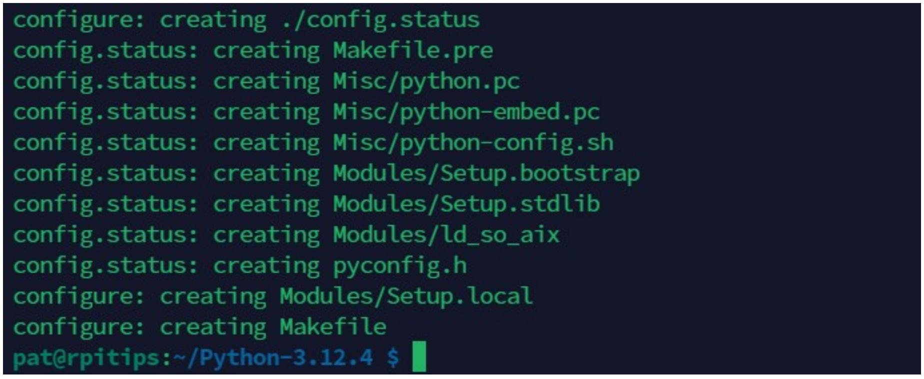
Related: [How To Use the Tar Command on Linux: A Complete Guide]

Are you a bit lost in the Linux command line? [Check this article first] for the most important commands to remember and a free downloadable cheat sheet so you can have the commands at your fingertips.

Configure and install Python latest version

Now we need to compile the source code to install this Python version on your Raspberry Pi:

- Move to the folder containing the extracted files:
  ```
  cd Python-3.12.4
  ```
- Run the configuration command:
  ```
  ./configure --enable-optimizations
  ```
  As Python is already installed on your Raspberry Pi, it should work directly. But if you have an error, you probably need to install or update the missing components (use [APT] to fix any error).



```
configure: creating ./config.status
config.status: creating Makefile.pre
config.status: creating Misc/python.pc
config.status: creating Misc/python-embed.pc
config.status: creating Misc/python-config.sh
config.status: creating Modules/Setup.bootstrap
config.status: creating Modules/Setup.stdlib
config.status: creating Modules/ld_so_aix
config.status: creating pyconfig.h
configure: creating Modules/Setup.local
configure: creating Makefile
pat@rpitips:~/Python-3.12.4 $
```

- Once done, run this command to install it:
  ```
  sudo make altinstall
  ```

This may take some time depending on your Raspberry Pi model and version (at least 15 minutes on Raspberry Pi 4/5 with Bookworm).

Make Python 3 the default version on Raspberry Pi OS

Each installed version of Python on your system adds a new executable in /usr/local/bin that you can use to run a program. For example, in my case I know have:

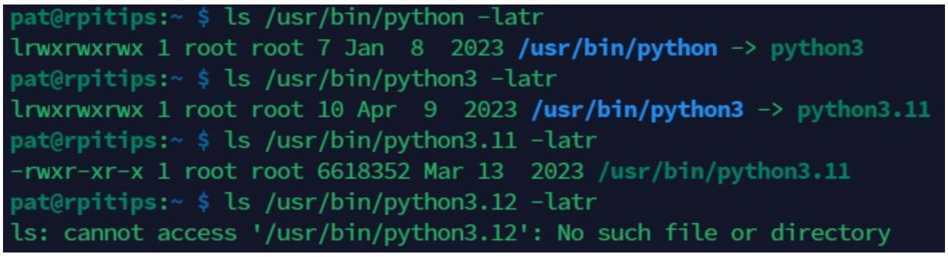
- python3 : The default Python 3 version.
- python3.11 : If I want to run a script with the version from the repository.
- python3.12 : The one I just installed from sources.

But when I use "python –version", I'm still using Python 3.11.
To choose the version you want to run, you have two choices:

- Always run a Python script specifying the exact version you want to use in the command line. For example:
  ```
  python3.12 myscript.py
  ```

Or, you can replace the link in your /usr/local/bin folder to point to the version you want to use by default (more details here).

Here is what it looks like on my Raspberry Pi currently:

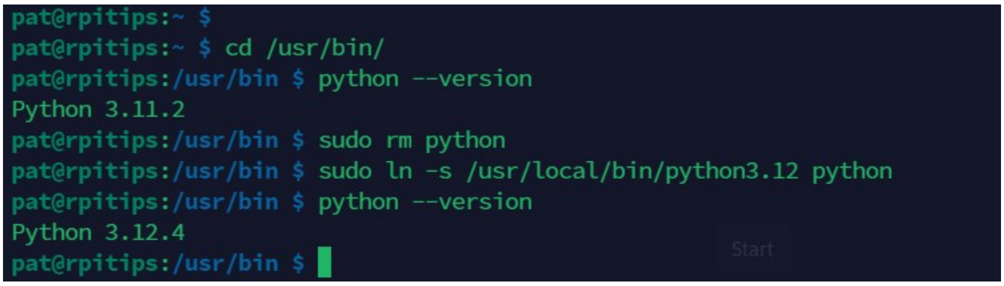


```
pat@rpitips:~ $ ls /usr/bin/python -latr
lrwxrwxrwx 1 root root 7 Jan  8  2023 /usr/bin/python -> python3
pat@rpitips:~ $ ls /usr/bin/python3 -latr
lrwxrwxrwx 1 root root 10 Apr  9  2023 /usr/bin/python3 -> python3.11
pat@rpitips:~ $ ls /usr/bin/python3.11 -latr
-rwxr-xr-x 1 root root 6618352 Mar 13  2023 /usr/bin/python3.11
pat@rpitips:~ $ ls /usr/bin/python3.12 -latr
ls: cannot access '/usr/bin/python3.12': No such file or directory
```

So basically, it will still use Python 3.11 no matter what command I use. Python 3.12 is available under /usr/local/bin, but not under /usr/bin yet.

And here is how to change this link to use the version you just installed:

- Go to /usr/bin:
  ```
  cd /usr/bin
  ```
- Remove the current link:
  ```
  sudo rm python
  ```
- Link the version you intend to use instead:
  ```
  sudo ln -s /usr/local/bin/python3.12 python
  ```
- Check that everything is fine:
  ```
  python --version
  ```
  It should now display the version you just installed (3.12.4 for me).



```
pat@rpitips:~ $
pat@rpitips:~ $ cd /usr/bin/
pat@rpitips:/usr/bin $ python --version
Python 3.11.2
pat@rpitips:/usr/bin $ sudo rm python
pat@rpitips:/usr/bin $ sudo ln -s /usr/local/bin/python3.12 python
pat@rpitips:/usr/bin $ python --version
Python 3.12.4
pat@rpitips:/usr/bin $ 
```

> **Note**: If you are using Thonny to code in Python, it uses "/usr/bin/python3" by default, which links to the latest version installed by Raspberry Pi OS (3.11.2 in my case). If you want to use the latest Python version with Thonny, you have to update this link as well.

Update Python on Raspberry Pi

To update Python on Raspberry Pi, start by making sure your whole system is up-to-date:

```
sudo apt update
sudo apt upgrade
```

Even if Raspberry Pi OS is always a few versions behind the latest Python version available, you can still get updates with apt, as for any other software on your device.

Once done, check the currently installed version with:

```
python --version
python3 --version
```

If it doesn't show the version you need to use, you will have to follow this tutorial from the beginning once again. Downloading the sources and compiling it for the desired version is the only solution each time you want to update, there isn't a magic command to do it automatically.

And it's important to know which version you are using when you install additional libraries for Python on your Raspberry Pi.

🛠️ This tutorial doesn't work anymore? Report the issue here, so that I can update it!

Want to chat with other Raspberry Pi enthusiasts? Join the community, share your current projects and ask for help directly in the forums.

You may also like:

- Programming your Raspberry Pi camera with Python
- Master Python on Raspberry Pi (ebook)
- Your Python script can have a GUI, here's how
- Managing Python libraries the right way
- Using GPIO pins with Python doesn't have to be complicated

Final thoughts

I hope this post made how the Python versions work on Raspberry Pi clear for you. You now know that there are at least two versions installed and that you can easily switch between them by changing the command you use.

Also, make sure to install the modules related to the Python version you use (python-gpiozero and python3-gpiozero are two different packages). PIP also has two versions (pip and pip3). It can be misleading for beginners, but it's really useful

Let me know in the community if you have any related question about Python on Raspberry Pi.
Meanwhile, feel free to check my other tutorials on Python on this website:

- How to Learn to Program in Python with a Raspberry Pi?
- How to program Minecraft with Python on a Raspberry Pi?
- How to Make a Discord Bot on Raspberry Pi? (Python easy method)

I also have a list of 15 projects with Python on Raspberry Pi that you can check if you need some inspiration.

Whenever you're ready, here are other ways I can help you:

The RaspberryTips Community: If you want to hang out with me and other Raspberry Pi fans, you can join the community. I share exclusive tutorials and behind-the-scenes content there. Premium members can also visit the website without ads.

Master your Raspberry Pi in 30 days: If you are looking for the best tips to become an expert on Raspberry Pi, this book is for you. Learn useful Linux skills and practice multiple projects with step-by-step guides.

The Raspberry Pi Bootcamp: Understand everything about the Raspberry Pi, stop searching for help all the time, and finally enjoy completing your projects.

Master Python on Raspberry Pi: Create, understand, and improve any Python script for your Raspberry Pi. Learn the essentials step-by-step without losing time understanding useless concepts.

You can also find all my recommendations for tools and hardware on this page.

How would you rate this article?

Click on a star to rate it!

★★★★★

Average rating 5 / 5. Vote count: 1

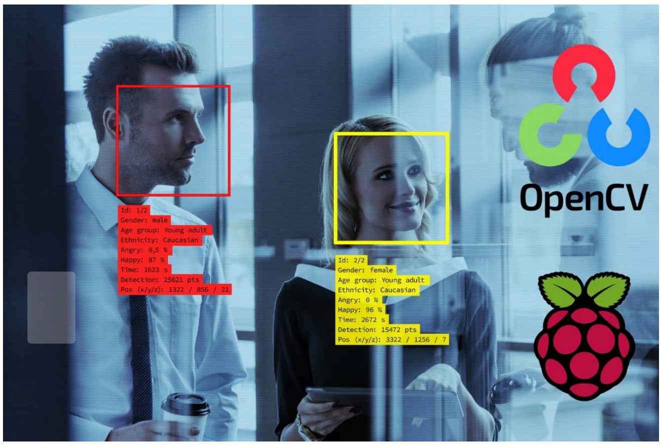
Patrick Fromaget

I'm the lead author and owner of RaspberryTips.com.
My goal is to help you with your Raspberry Pi problems using detailed guides and tutorials.
In real life, I'm a Linux system administrator with web developer experience.

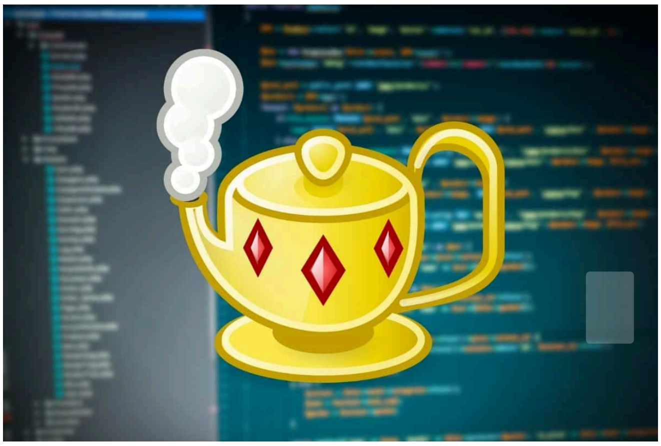
← PREVIOUS
How to Install Gentoo on Raspberry Pi: the Ultimate Guide

NEXT →
How to Download Files on Linux Using Terminal

Similar Posts



Install OpenCV on Raspberry Pi: The Only Guide You Need



How to Use Geany on Raspberry Pi? (Full guide with pictures)

5 Comments

Binyamin says:
June 1, 2022 at 3:43 pm

Thank you for buteaful tutorial
In my case I was still missing SSL module,
so I had to go back and install dependecies:

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

Jay k says:
June 15, 2022 at 11:00 am

Thanks for the guide it worked without errors but after following all the steps i got the error in pip. I used pip3.7 to look at all the libraries pre installed and also to install the libraries required. I got the following error

pi@raspberrypi:/ $ pip3.7 list
Package Version
———- ——-
pip   19.0.3
setuptools 40.8.0
Traceback (most recent call last):

```
  return command.main(cmd_args)
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/cli/base_command.py”, line 228, in main
  timeout=min(5, options.timeout)
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/cli/base_command.py”, line 93, in _build_session
  insecure_hosts=options.trusted_hosts,
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/download.py”, line 344, in __init__
  self.headers[“User-Agent”] = user_agent()
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/download.py”, line 108, in user_agent
  zip([“name”, “version”, “id”], distro.linux_distribution()),
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 120, in linux_distribution
  return _distro.linux_distribution(full_distribution_name)
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 675, in linux_distribution
  self.version(),
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 735, in version
  self.lsb_release_attr(‘release’),
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 892, in lsb_release_attr
  return self._lsb_release_info.get(attribute, ”)
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 550, in __get__
  ret = obj.__dict__[self._fname] = self._f(obj)
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 998, in _lsb_release_info
  stdout = subprocess.check_output(cmd, stderr=devnull)
 File “/usr/local/lib/python3.7/subprocess.py”, line 395, in check_output
  **kwargs).stdout
 File “/usr/local/lib/python3.7/subprocess.py”, line 487, in run
  output=stdout, stderr=stderr)
subprocess.CalledProcessError: Command ‘(‘lsb_release’, ‘-a’)’ returned non-zero exit status 1.
```

so what to do now please help!!!

Jay k says:
June 15, 2022 at 11:22 am

And this the error I'm getting if I try to install a library

```
pi@raspberrypi:/ $ pip3.7 install numpy
Exception:
Traceback (most recent call last):
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/cli/base_command.py”, line 179, in main
  status = self.run(options, args)
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/commands/install.py”, line 255, in run
  with self._build_session(options) as session:
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/cli/base_command.py”, line 93, in _build_session
  insecure_hosts=options.trusted_hosts,
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/download.py”, line 344, in __init__
  self.headers[“User-Agent”] = user_agent()
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/download.py”, line 108, in user_agent
  zip([“name”, “version”, “id”], distro.linux_distribution()),
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 120, in linux_distribution
  return _distro.linux_distribution(full_distribution_name)
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 675, in linux_distribution
```

```
  self.lsb_release_attr(‘release’),
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 892, in lsb_release_attr
  return self._lsb_release_info.get(attribute, ”)
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 550, in __get__
  ret = obj.__dict__[self._fname] = self._f(obj)
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 998, in _lsb_release_info
  stdout = subprocess.check_output(cmd, stderr=devnull)
 File “/usr/local/lib/python3.7/subprocess.py”, line 395, in check_output
  **kwargs).stdout
 File “/usr/local/lib/python3.7/subprocess.py”, line 487, in run
  output=stdout, stderr=stderr)
subprocess.CalledProcessError: Command ‘(‘lsb_release’, ‘-a’)’ returned non-zero exit status 1.
Traceback (most recent call last):
 File “/usr/local/bin/pip3.7”, line 10, in <module>
  sys.exit(main())
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/__init__.py”, line 78, in main
  return command.main(cmd_args)
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/cli/base_command.py”, line 228, in main
  timeout=min(5, options.timeout)
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/cli/base_command.py”, line 93, in _build_session
  insecure_hosts=options.trusted_hosts,
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/download.py”, line 344, in __init__
  self.headers[“User-Agent”] = user_agent()
 File “/usr/local/lib/python3.7/site-packages/pip/_internal/download.py”, line 108, in user_agent
  zip([“name”, “version”, “id”], distro.linux_distribution()),
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 120, in linux_distribution
  return _distro.linux_distribution(full_distribution_name)
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 675, in linux_distribution
  self.version(),
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 735, in version
  self.lsb_release_attr(‘release’),
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 892, in lsb_release_attr
  return self._lsb_release_info.get(attribute, ”)
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 550, in __get__
  ret = obj.__dict__[self._fname] = self._f(obj)
 File “/usr/local/lib/python3.7/site-packages/pip/_vendor/distro.py”, line 998, in _lsb_release_info
  stdout = subprocess.check_output(cmd, stderr=devnull)
 File “/usr/local/lib/python3.7/subprocess.py”, line 395, in check_output
  **kwargs).stdout
 File “/usr/local/lib/python3.7/subprocess.py”, line 487, in run
  output=stdout, stderr=stderr)
subprocess.CalledProcessError: Command ‘(‘lsb_release’, ‘-a’)’ returned non-zero exit status 1.
```

nephi says:
August 15, 2022 at 11:51 pm

followed your steps and I was able to download the current version 3.10.6 and get the file but I cant get the system to install that version it says im still using 3.7.3. ~/Python-3.10.6 $ python3 –version
Python 3.7.3

You can use python3.10 instead of python3.

Or update the symbolic link for python3, something like that:

```
cd /usr/bin/
sudo unlink python3
sudo ln -s /usr/bin/python3.10 python3
```

The easiest way is probably to uninstall Python 3.7 if you don't need it anymore.

Comments are closed.

Search

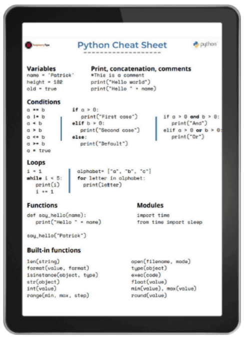
Welcome

RaspberryTips

Welcome! I'm Patrick – a Linux professional with a soft spot for the Raspberry Pi. Here, I share projects, tips, and lessons I've picked up on my journey. Whether you're a beginner or an enthusiast like me, I'm sure you'll find something helpful and inspiring here!

About RaspberryTips About me

My Python Cheat Sheet



Python Cheat Sheet

Variables

```
name = 'Patrick'
height = 182
old = true
```

Print, concatenation, comments

```
#This is a comment
print("Hello world")
print("Hello " + name)
```

Conditions

```
a == b
a != b
a < b
a > b
a <= b
a >= b
a = true
```

```
if a > 0:
  print("First case")
elif b > 0:
  print("Second case")
else:
  print("Default")
```

```
if a > 0 and b > 0:
  print("And")
elif a > 0 or b > 0:
  print("Or")
```

Loops

```
i = 1
while i < 5:
  print(i)
  i += 1
```

```
alphabet= ["a", "b", "c"]
for letter in alphabet:
  print(letter)
```

Functions

```
def say_hello(name):
  print("Hello " + name)

say_hello("Patrick")
```

Modules

```
import time
from time import sleep
```

Built-in functions

```
len(string)                    open(filename, mode)
format(value, format)          type(object)
isinstance(object, type)       exec(code)
str(object)                    float(value)
int(value)                     min(value), max(value)
range(min, max, step)          round(value)
```

Grab your one page summary of the Python syntax for your projects!

Free Download

Join The Community

- Chat with other Raspberry Pi users
- Ask for advices
- Get access to exclusive content
- Browse the website without ads