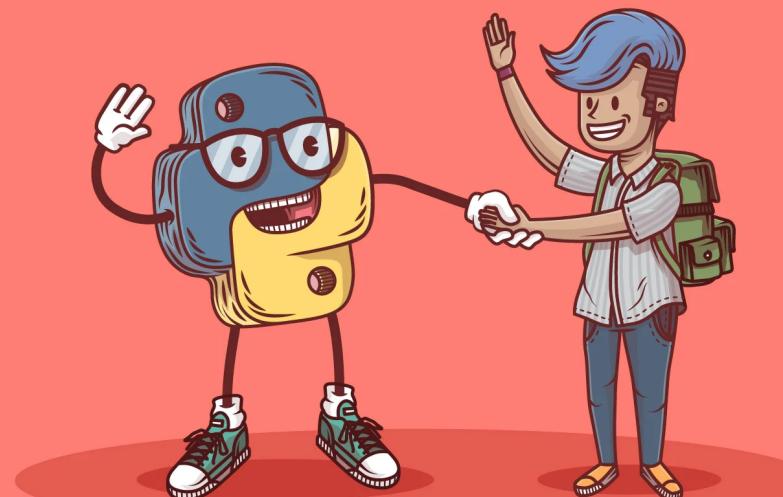


 Stuck at home? Enjoy free courses, on us →*Real Python*

Python 3 Installation & Setup Guide

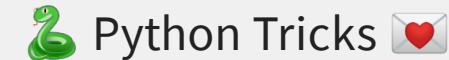
by Real Python  23 Comments  basics best-practices python

 Tweet  Share  Email

Table of Contents

- [Windows](#)
 - [Step 1: Download the Python 3 Installer](#)
 - [Step 2: Run the Installer](#)
- [Windows Subsystem for Linux \(WSL\)](#)

— FREE Email Series —



```
1 # How to merge two dicts
2 # in Python 3.5+
3
4 >>> x = {'a': 1, 'b': 2}
5 >>> y = {'b': 3, 'c': 4}
6
7 >>> z = {**x, **y}
8
9 >>> z
10 {'c': 4, 'a': 1, 'b': 3}
```

[Get Python Tricks »](#)

 No spam. Unsubscribe any time.

All Tutorial Topics

advanced api basics best-practices
community databases data-science
devops django docker flask front-end
intermediate machine-learning projects
python testing tools web-dev
web-scraping

- Linux
 - Ubuntu
 - Linux Mint
 - Debian
 - openSUSE
 - CentOS
 - Fedora
 - Arch Linux
 - Compiling Python From Source
- macOS / Mac OS X
 - Step 1: Install Homebrew (Part 1)
 - Step 2: Install Homebrew (Part 2)
 - Step 3: Install Python
- iOS (iPhone / iPad)
- Android (Phones & Tablets)
- Online Python Interpreters
- Conclusion



Your Guided Tour Through the Python Interpreter

Watch Now This tutorial has a related video course created by the author.

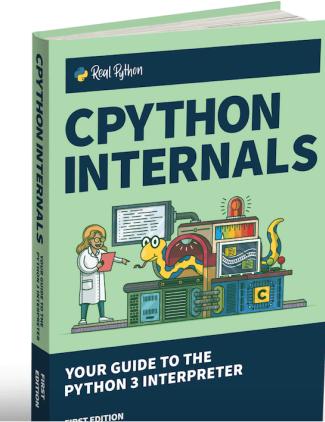
Watch it together with the written tutorial to deepen your understanding: [Installing Python on Windows, macOS, and Linux](#)

To get started working with Python 3, you'll need to have access to the Python interpreter. There are several common ways to accomplish this:

```

1 # How to merge two dicts
2 # in Python 3.5+
3
4 >>> x = {'a': 1, 'b': 2}
5 >>> y = {'b': 3, 'c': 4}
6
7 >>> z = {**x, **y}
8
9 >>> z
10 {'c': 4, 'a': 1, 'b': 3}

```



Improve Your Python

...with a fresh **Python Trick** ❤️ code snippet every couple of days:

Receive the Real Python newsletter and get notified about new tutorials we publish on the site, as well as occasional special offers.

Send Python Tricks »

Recommended Video Course

[Installing Python on Windows, macOS, and Linux](#)

Improve Your Python

X

^

- Python can be obtained from the **Python Software Foundation** website at [python.org](https://www.python.org). Typically, that involves downloading the appropriate **installer** for your operating system and running it on your machine.
- Some operating systems, notably Linux, provide a **package manager** that can be run to install Python.
- On macOS, the best way to install Python 3 involves installing a package manager called **Homebrew**. You'll see how to do this in the relevant section in the tutorial.
- On mobile operating systems like Android and iOS, you can install apps that provide a Python programming environment. This can be a great way to practice your coding skills on the go.



High Quality
Python Video Courses

Alternatively, there are several websites that allow you to access a Python interpreter online without installing anything on your computer at all.

Note: There is a chance that Python may have been shipped with your operating system and is already installed. Even if that is the case, it may be that the installed version is outdated, in which case you will want to obtain the latest version anyhow.

In this Python installation guide, you'll see step by step how to set up a working Python 3 distribution on Windows, macOS, Linux, iOS, and Android. So let's get started!

Free Download: Get a sample chapter from Python Basics: A Practical Introduction to Python 3 to see how you can go from beginner to intermediate in Python with a complete curriculum, up-to-date for Python 3.8.

Windows

It is highly unlikely that your Windows system shipped with Python already installed. Windows systems typically do not. Fortunately, installing does not involve much more than

downloading the Python installer from the [python.org website](https://python.org) and running it. Let's take a look at how to install Python 3 on Windows:

Step 1: Download the Python 3 Installer

1. Open a browser window and navigate to the [Download page for Windows](#) at python.org.
2. Underneath the heading at the top that says **Python Releases for Windows**, click on the link for the **Latest Python 3 Release - Python 3.x.x**. (As of this writing, the latest is Python 3.6.5.)
3. Scroll to the bottom and select either **Windows x86-64 executable installer** for 64-bit or **Windows x86 executable installer** for 32-bit. (See below.)

Sidebar: 32-bit or 64-bit Python?

For Windows, you can choose either the 32-bit or 64-bit installer. Here's what the difference between the two comes down to:

- If your system has a 32-bit processor, then you should choose the 32-bit installer.
- On a 64-bit system, either installer will actually work for most purposes. The 32-bit version will generally use less memory, but the 64-bit version performs better for applications with intensive computation.
- If you're unsure which version to pick, go with the 64-bit version.

Note: Remember that if you get this choice “wrong” and would like to switch to another version of Python, you can just uninstall Python and then re-install it by downloading another installer from python.org.

Step 2: Run the Installer

Once you have chosen and downloaded an installer, simply run it by double-clicking on the downloaded file. A dialog should appear that looks something like this:



Important: You want to be sure to check the box that says **Add Python 3.x to PATH** as shown to ensure that the interpreter will be placed in your execution path.

Then just click **Install Now**. That should be all there is to it. A few minutes later you should have a working Python 3 installation on your system.

Windows Subsystem for Linux (WSL)

If you are running Windows 10 Creators or Anniversary Update, you actually have another option for installing Python. These versions of Windows 10 include a feature called the **Windows Subsystem for Linux**, which allows you to run a Linux environment directly in Windows, unmodified and without the overhead of a virtual machine.

- For more information, see the [Windows Subsystem for Linux Documentation](#) article on the Microsoft website.
- For instructions on how to enable the subsystem in Windows 10 and install a Linux distribution, see the [Windows 10 Installation Guide](#).

- You can also check out this presentation on YouTube by [Sarah Cooley](#), one of the members of the WSL development team.

Once you have installed the Linux distribution of your choice, you can install Python 3 from a Bash console window, just as you would if you were running that Linux distribution natively. (See below.)

Linux

There is a very good chance your Linux distribution has Python installed already, but it probably won't be the latest version, and it may be Python 2 instead of Python 3.

To find out what version(s) you have, open a terminal window and try the following commands:

- `python --version`
- `python2 --version`
- `python3 --version`

One or more of these commands should respond with a version, as below:

Shell

```
$ python3 --versionhttp://feeds.megaphone.fm/lastarchive  
Python 3.6.5
```

If the version shown is Python 2.x.x or a version of Python 3 that is not the latest (3.6.5 as of this writing), then you will want to install the latest version. The procedure for doing this will depend on the Linux distribution you are running.

Note that it is frequently easier to use a tool called `pyenv` to manage multiple Python versions on Linux. To learn more about it, see our article [here](#).

Ubuntu

Depending on the version of the Ubuntu distribution you run, the Python install instructions vary. You can determine your local Ubuntu version by running the following command:

Shell

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.4 LTS
Release:        16.04
Codename:       xenial
```

Depending on the version number you see under Release in the console output, follow the instructions below:

- **Ubuntu 17.10, Ubuntu 18.04** (and above) come with Python 3.6 by default. You should be able to invoke it with the command python3.
- **Ubuntu 16.10 and 17.04** do not come with Python 3.6 by default, but it is in the Universe repository. You should be able to install it with the following commands:

Shell

```
$ sudo apt-get update
$ sudo apt-get install python3.6
```

You can then invoke it with the command python3.6.

- If you are using **Ubuntu 14.04 or 16.04**, Python 3.6 is not in the Universe repository, and you need to get it from a Personal Package Archive (PPA). For example, to install Python from the “[deadsnakes](#)” PPA, do the following:

Shell

```
$ sudo add-apt-repository ppa:deadsnakes/ppa  
$ sudo apt-get update  
$ sudo apt-get install python3.6
```

As above, invoke with the command `python3.6`.

Linux Mint

Mint and Ubuntu use the same package management system, which frequently makes life easier. You can follow the instructions above for **Ubuntu 14.04**. The “[deadsnakes](#)” PPA works with Mint.

Debian

We found sources that indicated that the Ubuntu 16.10 method would work for Debian, but we never found a path to get it to work on Debian 9. Instead, we ended up making Python from source as listed below.

One issue with Debian, however, is that it generally does not install the `sudo` command by default. To install it, you’ll need to do the following before you carry out the [Compiling Python From Source](#) instructions below:

Shell

```
$ su  
$ apt-get install sudo  
$ vi /etc/sudoers
```

After that, open the `/etc/sudoers` file using the `sudo vim` command (or your favorite text editor.) Add the following line of text to the end of the file, replacing `your_username` with your actual username:

Text

```
your_username ALL=(ALL) ALL
```

openSUSE

We found several sites describing how to get zypper to install the latest version of Python, but they seemed problematic or outdated. We did not manage to get any of them to work successfully, so we fell back to building Python from source. To do that, you will need to install the development tools, which can be done in YaST (via the menus) or by using zypper:

Shell

```
$ sudo zypper install -t pattern devel_C_C++
```

This step took a while and involved the installation of 154 packages, but once it was completed, we were able to build the source as shown in the [Compiling Python From Source](#) section above.

CentOS

The [IUS Community](#) does a nice job of providing newer versions of software for “Enterprise Linux” distros (i.e. Red Hat Enterprise and CentOS). You can use their work to help you install Python 3.

To install, you should first update your system with the [yum package manager](#):

Shell

```
$ sudo yum update  
$ sudo yum install yum-utils
```

You can then install the CentOS IUS package which will get you up to date with their site:

Shell

```
$ sudo yum install https://centos7.iuscommunity.org/ius-release.rpm
```

Finally you can then install Python and Pip:

Shell

```
$ sudo yum install python36u  
$ sudo yum install python36u-pip
```

Thanks to Jani Karhunen for his [excellent writeup](#) for CentOS 7.

Fedora

Fedora has a roadmap to switch to Python 3 as the default Python published [here](#). It indicates that the current version and the next few versions will all ship with Python 2 as the default, but Python 3 will be installed. If the python3 installed on your version is not 3.6, you can use the following command to install it:

Shell

```
$ sudo dnf install python36
```

Arch Linux

Arch Linux is fairly aggressive about keeping up with Python releases. It is likely you already have the latest version. If not, you can use this command:

Shell

```
$ packman -S python
```

Compiling Python From Source

Sometimes your Linux distribution will not have the latest version of Python, or maybe you just want to be able to build the latest, greatest version yourself. Here are the steps you need to take to build Python from source:

Step 1: Download the Source Code

To start, you need to get the Python source code. Python.org makes this fairly easy. If you go to the [Downloads](#) page, you will see the latest source for Python 3 at the top. (Make sure you don't grab Legacy Python, Python 2.)

When you select the version, at the bottom of the page there is a **Files** section. Select the **Gzipped source tarball** and download it to your machine. If you prefer a command line method, you can easily use wget to download it to your current directory:

Shell

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

Step 2: Prepare Your System

There are a few distro-specific steps involved in building Python from scratch. The goal of each step is the same on all distros, but you might need to translate to your distribution if it does not use apt-get:

1. The first step you should take when doing an operation like this is to update the system packages on your machine before you start. On Debian, this is what that looks like:

Shell

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

2. Next, we want to make sure the system has the tools needed to build Python. There are a bunch of them and you might already have some, but that's fine. I've listed them all in one command line, but you can break the list into shorter commands by just repeating the sudo apt-get install -y portion:

Shell

```
# For apt-based systems (like Debian, Ubuntu, and Mint)  
$ sudo apt-get install -y make build-essential libssl-dev zlib1g-dev libbz2-dev  
  
# For yum-based systems (like CentOS)  
$ sudo yum -y groupinstall development  
$ sudo yum -y install zlib-devel
```

Step 3: Build Python

1. Once you have the prerequisites and the tar file, you can unpack the source into a directory. Note that the following command will create a new directory called Python-3.6.5 under the one you are in:

Shell

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

2. Now you need to run the `./configure` tool to prepare the build:

Shell

```
$ ./configure --enable-optimizations --with-ensurepip=install
```

3. Next, you build the Python programs using `make`. The `-j` option simply tells `make` to split the building into parallel steps to speed up the compilation. Even with the parallel builds, this step can take a several minutes:

Shell

```
$ make -j 8
```

4. Then, you'll want to install your new version of Python. You'll use the `altinstall` target here in order to **not** overwrite the system's version of Python. Since you're installing Python into either `/usr/bin` or `/usr/local/bin`, you'll need to run as root:

Shell

```
$ sudo make altinstall
```

Warning: Please only use the `altinstall` target on `make`. Using the `install` target will overwrite the python binary. While this seems like it would be cool, there are big portions of the system that rely on the pre-installed version of Python.

Step 4: Verify Your Python Install

Finally, you can test out your new Python version:

```
$ python3.6 -V  
Python 3.6.5
```

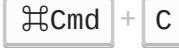
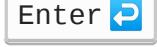
macOS / Mac OS X

While current versions of macOS (previously known as “Mac OS X”) include a version of Python 2, it is likely out of date by a few months. Also, this tutorial series uses Python 3, so let’s get you upgraded to that.

The best way we found to install Python 3 on macOS is through the [Homebrew package manager](#). This approach is also recommended by community guides like [The Hitchhiker’s Guide to Python](#).

Step 1: Install Homebrew (Part 1)

To get started, you first want to install Homebrew:

1. Open a browser and navigate to <http://brew.sh/>. After the page has finished loading, **select the Homebrew bootstrap code under “Install Homebrew”**. Then hit  to copy it to the clipboard. Make sure you’ve captured the text of the complete command because otherwise the installation will fail.
2. Now you need to **open a Terminal.app window, paste the Homebrew bootstrap code, and then hit **. This will begin the Homebrew installation.
3. If you’re doing this on a fresh install of macOS, you may get a pop up alert **asking you to install Apple’s “command line developer tools”**. You’ll need those to continue with the installation, so please **confirm the dialog box by clicking on “Install”**.

At this point, you’re likely waiting for the command line developer tools to finish installing, and that’s going to take a few minutes. Time to grab a coffee or tea!

Step 2: Install Homebrew (Part 2)

You can continue installing Homebrew and then Python after the command line developer tools installation is complete:

1. Confirm the “The software was installed” dialog from the developer tools installer.
2. Back in the terminal, hit **Enter ↵** to continue with the Homebrew installation.
3. Homebrew asks you to enter your password so it can finalize the installation. **Enter your user account password and hit Enter ↵** to continue.
4. Depending on your internet connection, Homebrew will take a few minutes to download its required files. Once the installation is complete, you’ll end up back at the command prompt in your terminal window.

Whew! Now that the Homebrew package manager is set up, let’s continue on with installing Python 3 on your system.

Step 3: Install Python

Once Homebrew has finished installing, **return to your terminal and run the following command:**

Shell

```
$ brew install python3
```

Note: When you copy this command, be sure you don’t include the \$ character at the beginning. That’s just an indicator that this is a console command.

This will download and install the latest version of Python. After the Homebrew brew install command finishes, Python 3 should be installed on your system.

You can make sure everything went correctly by testing if Python can be accessed from the terminal:

1. Open the terminal by launching **Terminal.app**.
2. Type pip3 and hit **Enter ↵**.
3. You should see the help text from Python’s “Pip” package manager. If you get an error message running pip3, go through the Python install steps again to make sure you have a working Python installation.

Assuming everything went well and you saw the output from [Pip](#) in your command prompt window...congratulations! You just installed Python on your system, and you’re all set to continue with the next section in this tutorial.

iOS (iPhone / iPad)

The [Pythonista app](#) for iOS is a full-fledged Python development environment that you can run on your iPhone or iPad. It’s basically a combination of a Python editor, documentation, and interpreter rolled into one single app.

Pythonista is surprisingly fun to use. It’s a great little tool when you’re stuck without a laptop and want to work on your Python skills on the go. It comes with the complete Python 3 standard library and even includes full documentation you can browse offline.

To install and set up Pythonista you need to [download it from the iOS app store](#).

Android (Phones & Tablets)

If you have an Android tablet or phone and want to practice Python on the go, there are several options available. The one that we found most reliably supports Python 3.6 is [Pydroid 3](#).

Pydroid 3 features an interpreter you can use for REPL sessions, and it also provides the ability to edit, save, and execute Python code:

You can [download and install Pydroid 3 from the Google Play store](#). There is a free version and also a paid Premium version which supports code prediction and code analysis.

Online Python Interpreters

If you want to try out the examples in this tutorial without installing Python on your machine, there are several web sites available where you can interact with a Python interpreter online:

- Python.org Online Console: www.python.org/shell
- Python Fiddle: pythonfiddle.com
- Repl.it: repl.it
- Trinket: trinket.io
- Python Anywhere: www.pythonanywhere.com

These cloud-based Python interpreters may not be able to execute some of the more complex examples in this tutorial, but they will be adequate for running most of the simpler ones and may be a nice way to get you started. More information on using these sites is presented in the next section.

Conclusion

This section provided you with the information you need to gain access to a Python 3 interpreter. You are now ready to head to the next section and begin interacting with Python!

[« Introduction to Python](#)

[Installing Python](#)

[Interacting with Python »](#)

 **Watch Now** This tutorial has a related video course created by the Real Python team. Watch it together with the written tutorial to deepen your understanding: [Installing Python on Windows, macOS, and Linux](#)



Python Tricks 

Get a short & sweet **Python Trick** delivered to your inbox every couple of days. No spam ever. Unsubscribe any time. Curated by the Real Python team.

```
1 # How to merge two dicts
2 # in Python 3.5+
3
4 >>> x = {'a': 1, 'b': 2}
5 >>> y = {'b': 3, 'c': 4}
6
7 >>> z = {**x, **y}
8
9 >>> z
10 {'c': 4, 'a': 1, 'b': 3}
```

Email Address

[Send Me Python Tricks »](#)

About The Team

Each tutorial at Real Python is created by a team of developers so that it meets our high quality standards. The team members who worked on this tutorial are:



Dan



Jim

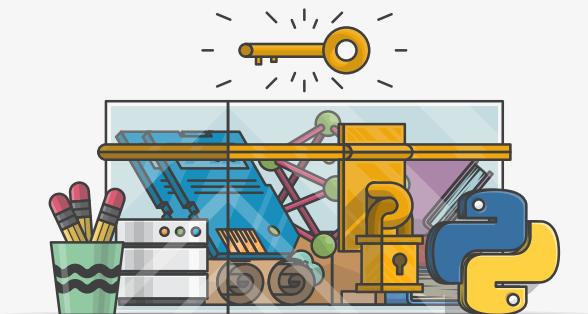


Joanna



John

Master Real-World Python Skills
With Unlimited Access to Real Python



Join us and get access to hundreds of tutorials, hands-on video courses, and a community of expert Pythonistas:

[Level Up Your Python Skills »](#)

What Do You Think?

[Tweet](#)

[Share](#)

[Email](#)

Real Python Comment Policy: The most useful comments are those written with the goal of learning from or helping out other readers—after reading the whole article and all the earlier comments. Complaints and insults generally won't make the cut here.

What's your #1 takeaway or favorite thing you learned? How are you going to put your newfound skills to use? Leave a comment below and let us know.

Keep Learning

Related Tutorial Categories: [basics](#) [best-practices](#) [python](#)

Recommended Video Course: [Installing Python on Windows, macOS, and Linux](#)

© 2012–2020 Real Python · [Newsletter](#) · [Podcast](#) · [YouTube](#) · [Twitter](#) · [Facebook](#) · [Instagram](#) ·
[Python Tutorials](#) · [Search](#) · [Privacy Policy](#) · [Energy Policy](#) · [Advertise](#) · [Contact](#)

 Happy Pythoning!