



Local Python: Installation Directions

Python 3

You probably recall from the pre-work that both Python 2 and Python 3 are currently in use. Python 3 was released in 2008 and included breaking changes. This means that an upgrade changes the way some code works, and old code won't always be compatible with the way new code wants it to be written.

Why did Python 3 introduce [breaking changes](#)? It turns out there were some things in Python 2 that could have been designed better. [Guido](#) and the Python community decided that it would be worth it in the long haul to fix those mistakes and get the language back on track for where they want it to be in the future. Upgrading to Python 3 is a good thing!

Why are some people still using Python 2?

It's easy to convert small projects from Python 2 to Python 3, but large, complex projects are a challenge. In an ideal world, everyone could convert their projects and Python 2 would be laid to rest. In reality, lots of large, popular projects began in Python 2 and the creators don't have the resources (or incentives) to upgrade.

As a reminder, **all new projects** should be started in Python 3! Everything we do in this course will always be in Python 3.

Installing Python 3

Instructions vary slightly depending on what kind of machine you're using. Click the link below that applies to you:

[Installation Instructions: Mac](#)

[Installation Instructions: Linux](#)

[Installation Instructions: Windows](#)

Installation Instructions: Mac

Macs usually come with Python 2 already installed. We're going to run through some installation steps to make sure you've got the latest and greatest that Python has to offer.

1. Open up your terminal.

You can do this by pressing command+space bar and typing "terminal," or by locating the application and clicking on the icon.

2. Install XCode with the following command.

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

This may take a few minutes. Once it's done, you can run the following command to make sure it's installed properly.

```
xcode-select -p
```

Installation Instructions: Mac

Your output should look something like this:

```
/Applications/Xcode.app/Contents/Developer
```

3. Install `Homebrew` by running the following command.

Pro tip: Do not try to type this in. Copy and paste to make sure everything is correct. Do this by selecting the text with your cursor and pressing `command+C`. Then, go to your terminal and press `command+V`.

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master
```

Once this command runs, type `brew doctor` on your terminal prompt. If you get the output `Your system is ready to brew`, you are ready to move on to the next step.

Installation Instructions: Mac

4. Add PATH environment variable.

This is a bit confusing, but basically we're setting the path up so Homebrew knows where to install something.

```
open ~/.profile
```

The file should open up. Ask your instructor for help if it didn't. Copy and paste the following line at the bottom of this file:

```
export PATH=/usr/local/bin:/usr/local/sbin:$PATH
```

Save the changes and close the file.

Installation Instructions: Mac

5. Install Python 3 (finally!).

Homebrew, by default, gets the latest stable version of whatever you're trying to install.

```
brew install python
```

6. Create an alias for `python3`.

```
open ~/.bashrc
```

At the bottom of that file, copy and paste the following lines:

```
alias python=python3  
alias pip=pip3
```

Learn more about aliases [here](#).

Installation Instructions: Mac

7. Restart your Terminal.

Right click (control+click on most Macs) on the Terminal icon in your application tray. Select `Quit` from the menu to make sure Terminal is fully stopped. Then, open it again (see Step 1).

***Pro tip:** Your settings won't be updated until Terminal is fully stopped and restarted. If you simply minimize the program, you will not see any updates!*

8. Check version.

```
python --version
```

You will get something like this. As long as it starts with a 3, you're good to go!

Python 3.6.5

Now let's check `pip`, the package installer.

```
pip --version
```

Installation Instructions: Mac

9. Let's put `pip` to the test!

Try installing `ipython` with `pip`. `ipython` is an environment that you can use to run a REPL right from your terminal, so it can be useful to have.

```
pip install ipython
```

Installation Instructions: Mac

10. (Bonus, if you're waiting for others to finish.)

Type the following command in your terminal:

```
ipython
```

You are now in a development environment! [Click here](#) for more instructions.

Installation Instructions: Linux

Pro tip: The instructions are for Ubuntu. If you have another version of Linux, please follow these [suggested directions](#).

1. Open your terminal.

Either:

- Click Ubuntu icon (upper-left corner) to open Dash. Then, type “terminal” and select Terminal from the results.

Or:

- Hit the keyboard shortcut `Ctrl - Alt + T`.

Installation Instructions: Linux

2. Check to see if Python 3 exists.

Some distributions of Linux come with Python 3 already installed. How nice! To check if you have Python 3 already, run the following command:

```
python3 --version
```

If it gives you a version, you're good to go! Go ahead and skip the rest of the directions and go to the [bonus section](#) instead. Otherwise, move to Step 3.

Installation Instructions: Linux

3. Install Python 3.6.

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

Check again for the Python 3 version.

```
python3 --version
```

This time, things should be all good. Go ahead and jump down to the [bonus section](#)!

If you are still unable to get Python 3, please alert your instructor now.

Installation Instructions: Windows

Pro tip: If you have Windows XP, you need to be downgraded from Python 3.6 to 3.4. Please ask your instructor for help if you plan on using Windows XP.

1. Download the Python installer.

Visit python.org and download the web-based installer for Windows. You'll find this under a “Files” section at the bottom of the page.

If you have 64-bit Windows, use the link that contains **64**. If you have 32-bit Windows, download the one without **64**. If you have no idea what you have, [click here to learn how to find out](#).

Installation Instructions: Windows

2. Run the installer.

- Make sure both `Add Python 3.6 to PATH` and `Install for all users` are checked.
- Click `Install Now`.

Installation Instructions: Windows

3. Disable length limit.

After the initial installation is finished, there will be an additional option that says something about a max character limit. **You want this!** Provide permission for this setting to be changed.

4. Open your terminal.

- Click *Start*.
- Open *Windows System* menu.
- Select *Command Prompt*.

Installation Instructions: Windows

5. Run the `py` command.

```
py
```

You should get a message telling you what version of Python you're using as well as opening an in-terminal REPL. If you did, great! Skip to the next step.

If you instead received an error message like the one below, something went wrong and Python didn't install correctly.

```
'py' is not recognized as an internal or external command,  
operable program or batch file.
```

In this case, ask your instructor for assistance.

Installation Instructions: Windows

6. (Bonus, if you are waiting for others to finish.)

[Click here.](#)

Windows 64-Bit or 32-Bit

Pro tip: These directions are for Windows 7 and Windows Vista operating systems. If you have Windows 10, you most likely have a 64-bit machine, but if you want to be extra sure, [check here](#).

1. Open “System” by clicking the “Start” button.
2. Right click “Computer.”
3. Click “Properties.”
4. Under “System,” you can view the system type.

This will give you a bunch of stats about your machine, including whether it is 32-bit or 64-bit.

5. Return to [Installation Instructions: Windows](#).

Bonus

This is an optional section, in case you are waiting for others to finish.

Let's test out our in-line REPL! You can type and execute code one line at a time. Let's do a simple exercise from our earlier variables lesson:

```
x = 10 print(x)
```

Compared to using a text editor or the online repl.it, how does `ipython` stack up? What do you think it's useful for?

- Play around a little bit with some mathematical operators:
 - `+`, `-`, `*`, `/`, and `**`.
- Try exiting (`exit` command) and restarting your in-line REPL.
 - Does it still remember `x` from before?
- Can you print strings as well?
 - Try concatenating multiple strings together.

Still waiting? Check out this video about [what you can do with Python!](#)

Additional Resources

- [List of Differences Between Python 2 and 3](#)
- [Python 2 or 3?](#)
- [Official OSX Installation Instructions](#)
- [Official Windows Installation Instructions](#)
- [Windows-Specific Modules](#)