



Installing Python The IRE Way™

Updated 2019-02-21

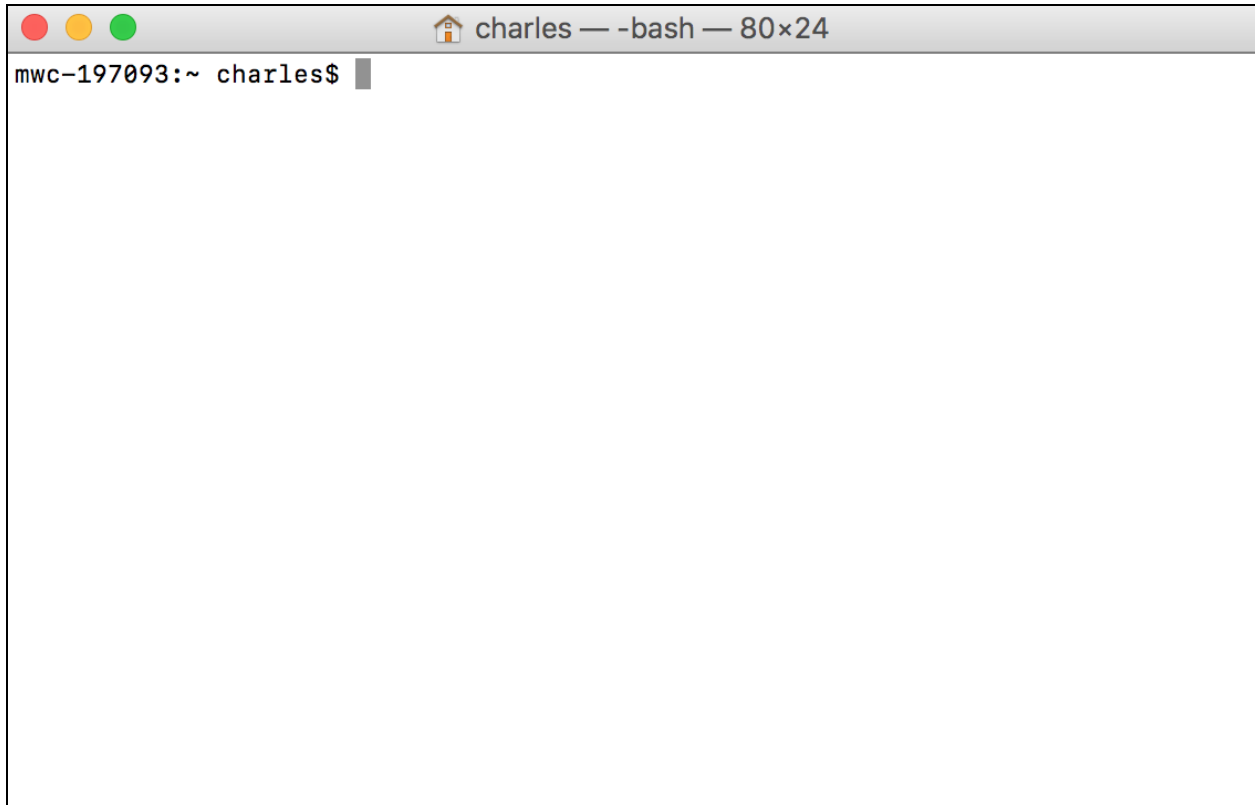
This is an opinionated guide to installing Python and a tool called [pipenv](#) to manage your Python projects. We'll generally follow the installation guides found in [A Hitchhiker's Guide To Python](#). You'll need an internet connection and a little bit of time.

Questions? Comments? Problems? Email training@ire.org.

- [Installing on a Mac](#)
- [Installing on a PC](#)
- [Setting up a new project](#)
- [Working on an existing project](#)

Installing on a Mac

Open your Terminal app. It should be in Applications → Utilities, or you can spotlight search for "Terminal." When it opens, you should see something like this:



First, install xcode's command-line tools (we need the code compiler).
Paste this code into your terminal and hit enter:

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

This will take a minute.

If you don't have it already, you'll want to install [homebrew](#), a handy package manager for OSX. Good for lots of stuff besides Python.

To install Homebrew, paste this code into your terminal and hit enter:

```
/usr/bin/ruby -e "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/i  
ninstall)"
```

This could take a minute.

Next, add the homebrew path to your \$PATH environment variable.

If you have OSX 10.12 (Sierra) or older, paste this code into your terminal and hit enter:

```
echo "export PATH=/usr/local/bin:/usr/local/sbin:$PATH" >>
~/.profile
```

Otherwise, paste this code into your terminal and hit enter:

```
echo "export PATH="/usr/local/opt/python/libexec/bin:$PATH"
" >> ~/.profile
```

Now (finally!): Install Python3 and pipenv with brew. In your Terminal:

```
brew install python3 pipenv
```

When that finishes, double check that Python3 is there:

```
python3 --version
```

That should return a version number, not an error. Let's also check on pipenv:

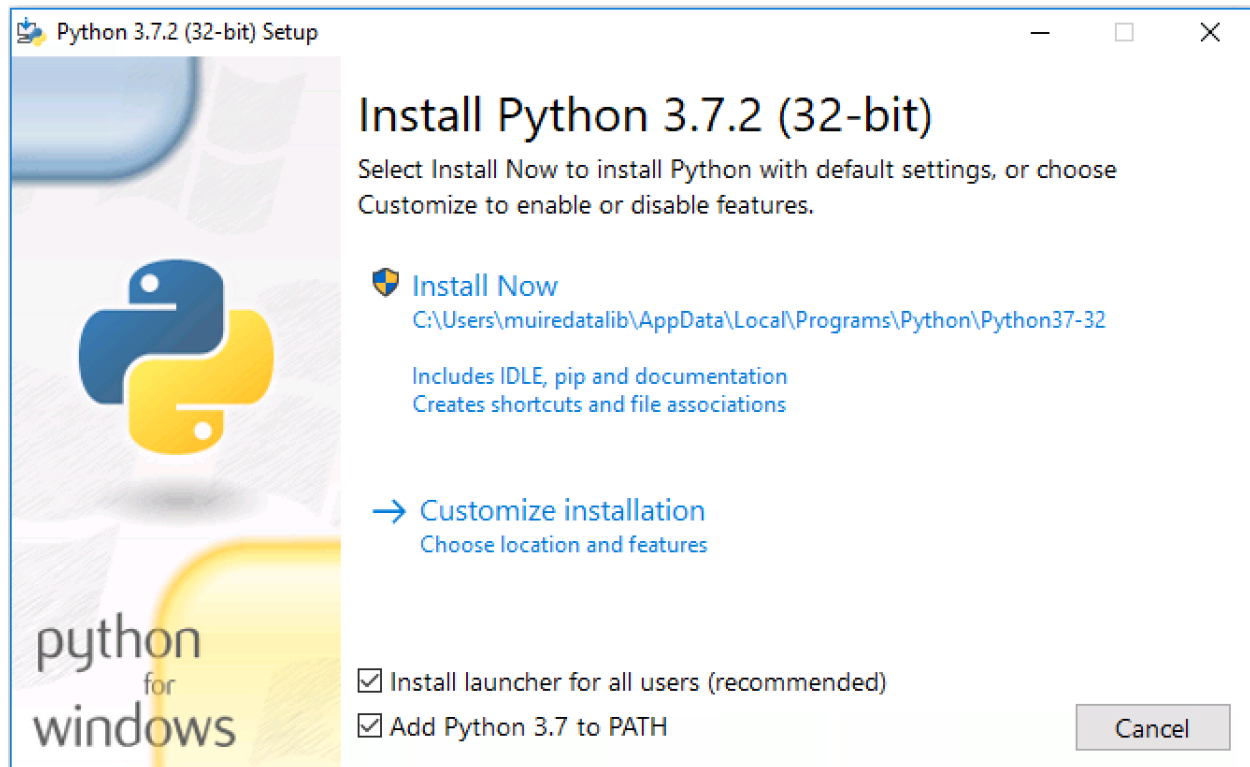
```
pipenv --version
```

If you get back a version number and not an error, great! You're ready to [get started on your first project](#).

Installing on a PC

First, [download the Windows installer](#) for the most recent version of Python 3 (Python 3.7.2, as of this writing) and run it.

Important: Make sure that the "Install launcher for all users" and the "Add Python 3.7 to PATH" checkboxes at the bottom are both checked.



Next, open a command prompt window. The quickest way to do this is to search for a program called *cmd*. Should look something like this:

```
cmd Command Prompt - cmd
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\markc>
```

Double-check that Python is installed and available from the command line:

```
python --version
```

... which should return a version number, not an error. If you get an error message that says "python is not recognized as an internal or external

command,” uninstall and reinstall Python, making sure that you checked the “Add Python to PATH” checkbox.

If you’re still having problems, it’s probably a dumb PATH issue -- contact training@ire.org and we’ll help you get it sorted out.

Next step, install *pipenv*:

```
pip install --user pipenv
```

If you get a message that says “pip is not recognized as an internal or external command,” uninstall and reinstall, as above.

Close out of your command prompt and open a new one. Check to see if pipenv installed correctly:

```
pipenv --version
```

... which should return a version number, not an error. If you get an error (there’s a common one that pops up here), contact us and we’ll walk you through fixing it.

Setting up a new project

Open your terminal or command prompt and create a new folder to hold your project files (make your directory name lowercase with no spaces):

```
mkdir your-cool-project
```

Then move into your new directory:

```
cd your-cool-project
```

Use `pipenv` to install any third-party modules you'll need. If you wanted to use jupyter notebooks and you're working on a scraper, for instance, you might do something like:

```
pipenv install jupyter requests bs4
```

Now, to launch your jupyter notebook server, you would run this command:

```
pipenv run jupyter notebook
```

Working on an existing project

If you want to work on an existing project -- say, some Python in a GitHub repo that you saw at a conference that you've copied onto your computer -- you would follow these steps, assuming the project came with a *Pipfile* or a *requirements.txt* file included. (If not, you'll have to install any third-party dependencies manually, as above.)

For example, again, let's say you're working on a project that uses jupyter notebooks.

First, move into the directory:

```
cd some-cool-project
```

I usually run `ls` (if you're on a PC, the command is *dir*) to list all the files in the directory to make sure everything's there -- again, you're looking for a *Pipfile* or a *requirements.txt*.

```
ls
```

Assuming you have what you need, install the third-party packages you need:

```
pipenv install
```

Once that finishes, you can run the notebook server:

```
pipenv run jupyter notebook
```