

Learn Python: pipenv

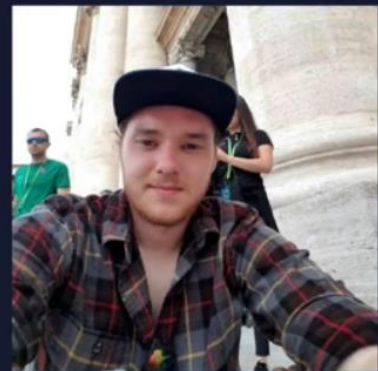
In this article, you'll learn how to install pipenv on Windows, MacOS, and Chromebook computers.

Installing `pipenv` on MacOS and Windows

Python Walkthrough

Virtual Environments w/ `pipenv`

with Mike Dane



`code`cademy

Key Concepts

This video will cover the following key concepts:

- What are virtual environments and why are they useful
- Installing pipenv (Windows & Mac)
- Working with virtual environments

`code`cademy

numpy · PyPI requests · PyPI

Python Software Foundation [US] | <https://pypi.org/project/requests/#history>

Search projects

Help Donate Log in Register

requests 2.20.1

✓ Latest version

`pip install requests`

Last released: Nov 8, 2018

Python HTTP for Humans.

Navigation

Project description

Release history

Download files

Release history

Release notifications

THIS VERSION

2.20.1
Nov 8, 2018

2.20.0

numpy · PyPI requests · PyPI

Python Software Foundation [US] | <https://pypi.org/project/requests/#history>

(Apache 2.0)

Author: Kenneth Reitz

Requires: Python >=2.7, !=3.0.*, !=3.1.*, !=3.2.*, !=3.3.*

Maintainers

graffatcolmingov

Lukasa

nateprewitt

Classifiers

Development Status
5 - Production/Stable

Intended Audience
Developers

License
OSI Approved :: Apache Software

2.18.4
Aug 15, 2017

2.18.3
Aug 2, 2017

2.18.2
Jul 25, 2017

2.18.1
Jun 14, 2017

2.18.0
Jun 14, 2017

2.17.3

<https://pypi.org/project/requests/2.18.2/>

The top screenshot shows the NumPy project page on PyPI. The version 1.15.4 is highlighted, and the command `pip install numpy` is displayed. The page also shows the project description: "NumPy: array processing for numbers, strings, records, and objects." and the release date: "Last released: Nov 4, 2018".

The bottom screenshot shows the "Release history" section, listing the following versions and release dates:

- 1.15.4 (Nov 4, 2018) - THIS VERSION
- 1.15.3 (Oct 22, 2018)
- 1.15.2 (Sep 23, 2018)
- 1.15.1 (Aug 21, 2018)
- 1.15.0 (Jul 23, 2018)
- 1.15.0rc2 (Jul 9, 2018) - PRE-RELEASE
- 1.15.0rc1 (Jun 21, 2018) - PRE-RELEASE

Installing `pipenv` on MacOS

1. First, let's check that we have `pip` using the `pip3 --version` command. If you're using Python 2, you'll use the `pip --version` command instead.

```
My-Mac:~ codecademy$ pip3 --version
pip 18.1 from /usr/lib/python3/dist-packages/pip (python 3.7)
```

If you instead see:

```
-bash: pip3: command not found
```

you may need to update or reinstall Python. [This article about Installing Python 3 and Python Packages can help.](#)

2. Next, let's install `pipenv` using the `pip3 install --user pipenv` command:

```
My-Mac:~ codecademy$ pip3 install --user pipenv
You may see some warnings about certain directories not being on PATH. This
means, if we try the pipenv command, it might not work!

The scripts pipenv and pipenv-resolver are installed in
'/home/yourusername/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use
--no-warn-script-location.

...

$ pipenv
-bash: pipenv: command not found
Let's fix that!
```

3. We may need to add the directory `pipenv` is installed in to your `PATH`. We may need to edit our `~/.bash_profile` file using the `vi` editor in our terminal. **If you find yourself getting confused using `vi`, watch the video above to see someone use `vi`.**

```
My-Mac:~ codecademy$ vi ~/.bash_profile
This will open a file with some code already in it! Check for the lines:
```

```
# set PATH so it includes the user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
export PATH="$HOME/.local/bin:$PATH"
fi
```

If your file has these lines, you're good to go! Skip to the end of this step where we save and exit the file. If you don't see those lines you will need to add them to your file.

Press the `⏎` key to enter `INSERT` mode which allows you to type in the file.

At the bottom of the file, add the lines:

```
# set PATH so it includes the user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
export PATH="$HOME/.local/bin:$PATH"
fi
```

Then, we need to save and exit the file. To do this, we need to:

- Press the `esc` key to exit `INSERT` mode
- Type `:` which will allow us to enter a `vi` command
- Press the `w` key (to save the file), the `q` key (to exit the file), and `⏎` to force the command

If this is working correctly, the bottom of the file should look like:

```
if [ -d "$HOME/.local/bin" ] ; then
PATH="$HOME/.local/bin:$PATH"
fi
~
~
:wq!
```

Now, press the `Enter` key.

Note: If you don't see the `:` before the `wq!` this means you're typing the letters into the file instead of using a `vi` command. Erase the letters and try pressing the `esc` key to exit **INSERT** mode again.

4. Next, we'll use the command `source ~/.bash_profile` to load these environment variables into the current shell.

```
My-Mac:~ codecademy$ source ~/.bash_profile
```

Now, typing `pipenv --version` should work!

```
My-Mac:~ codecademy$ pipenv --version
pipenv, version 2021.5.29
```

Installing `pipenv` on Windows



```
Command Prompt - pip install --user pipenv
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\miked>pip --version
pip 18.1 from c:\users\miked\appdata\local\programs\python\python37-32\lib\site-packages\pip (python 3.7)

C:\Users\miked>pip install --user pipenv
Collecting pipenv
  Using cached https://files.pythonhosted.org/packages/7d/ed/526ea0c23f95cc431e67768718de91a69083c060118c5fcc791e8f23141/pipenv-2018.11.14-py3-none-any.whl
```

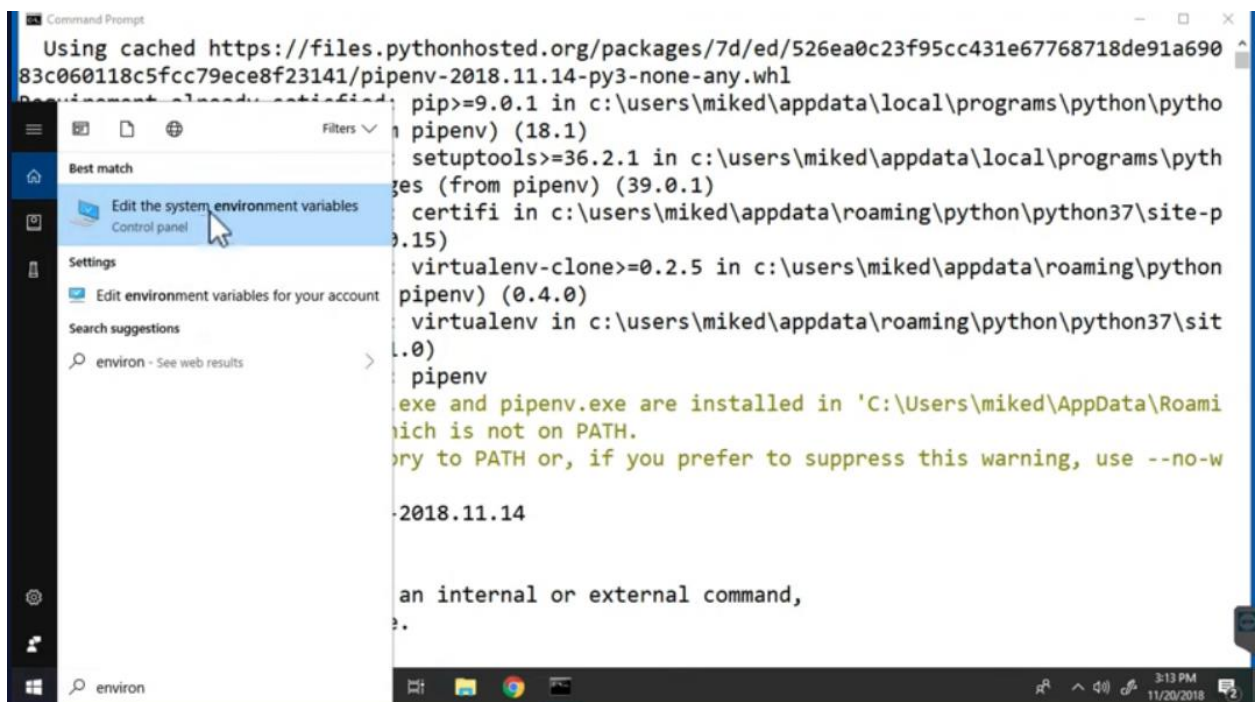
```
Select Command Prompt
python 3.7)

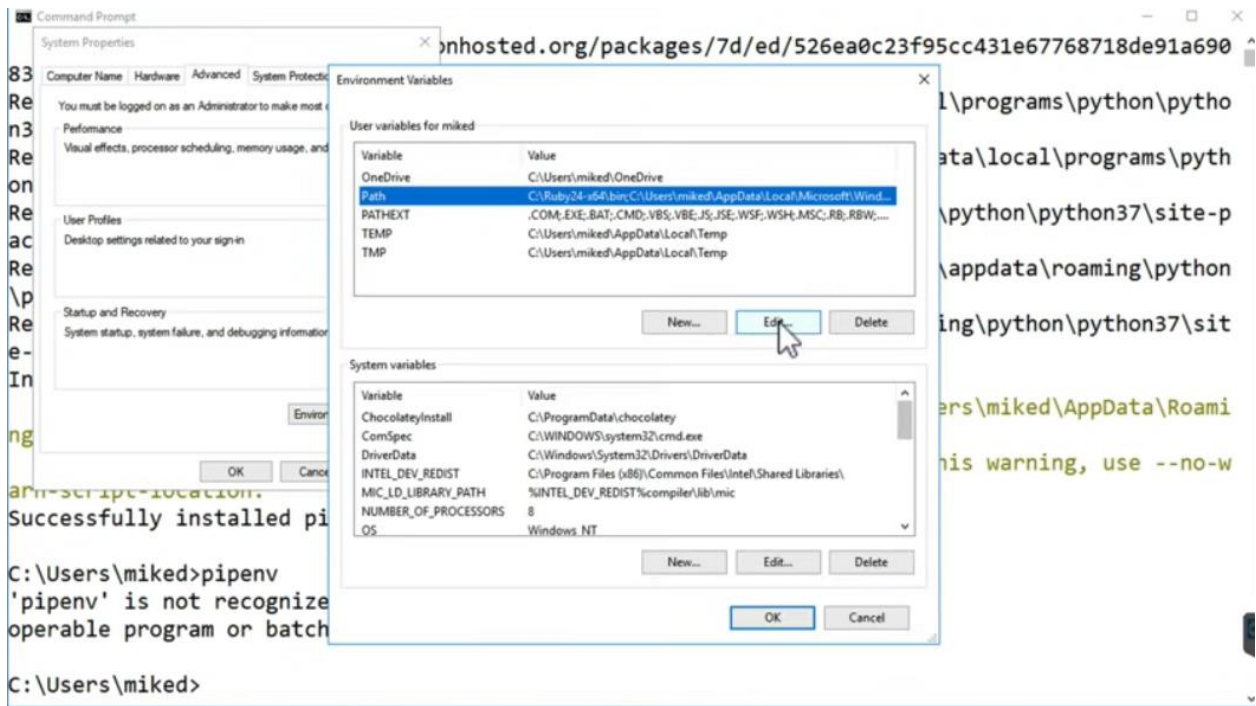
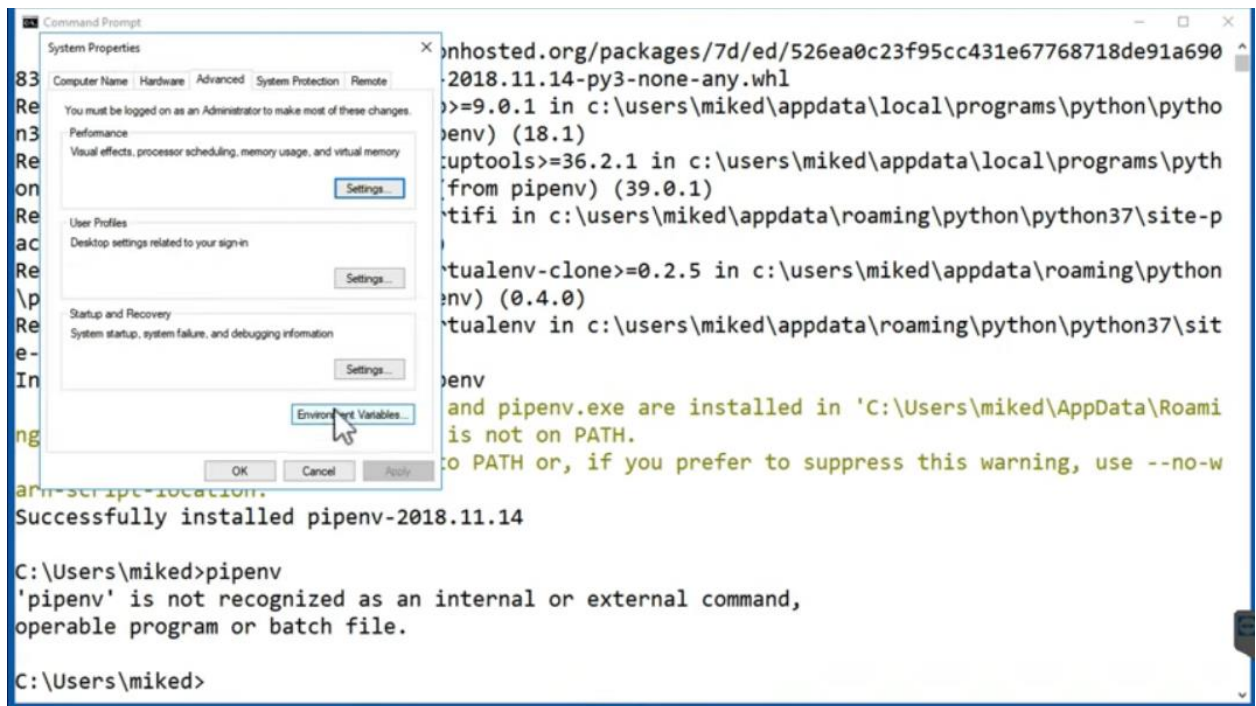
C:\Users\miked>pip install --user pipenv
Collecting pipenv
  Using cached https://files.pythonhosted.org/packages/7d/ed/526ea0c23f95cc431e67768718de91a69083c060118c5fcc79ece8f23141/pipenv-2018.11.14-py3-none-any.whl
Requirement already satisfied: pip>=9.0.1 in c:\users\miked\appdata\local\programs\python\python37-32\lib\site-packages (from pipenv) (18.1)
Requirement already satisfied: setuptools>=36.2.1 in c:\users\miked\appdata\local\programs\python\python37-32\lib\site-packages (from pipenv) (39.0.1)
Requirement already satisfied: certifi in c:\users\miked\appdata\roaming\python\python37\site-packages (from pipenv) (2018.10.15)
Requirement already satisfied: virtualenv-clone>=0.2.5 in c:\users\miked\appdata\roaming\python\python37\site-packages (from pipenv) (0.4.0)
Requirement already satisfied: virtualenv in c:\users\miked\appdata\roaming\python\python37\site-packages (from pipenv) (16.1.0)
Installing collected packages: pipenv
  The scripts pipenv-resolver.exe and pipenv.exe are installed in 'C:\Users\miked\AppData\Roaming\Python\Python37\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pipenv-2018.11.14

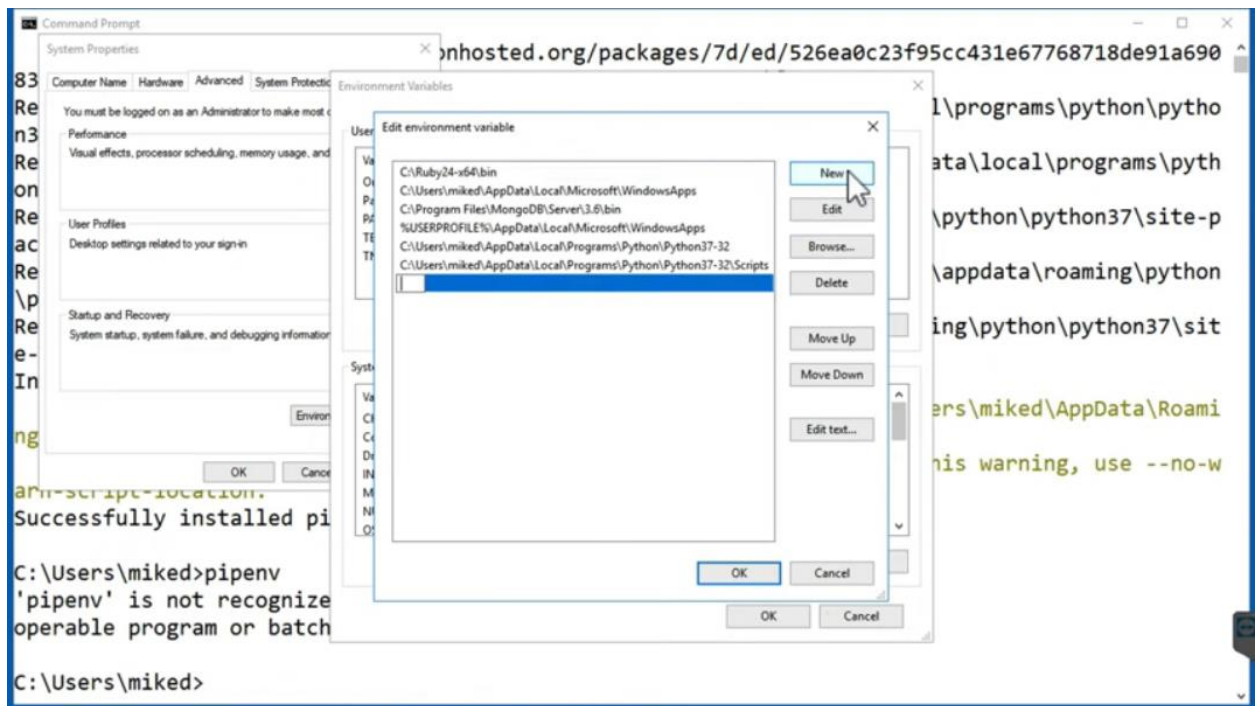
C:\Users\miked>
```

We see a warning: The scripts pipenv-resolver.exe and pipenv.exe are installed in which is not on PATH.

PATH is a system variable that Windows uses in order to execute where the executables are.







C:\Users\Andres R. Bucheli\AppData\Roaming\Python\Python39\Scripts



Installing `pipenv` on Windows

1. Open the Command Prompt by typing "cmd" in the Windows Search Bar.
2. Now, let's check that we have `pip` using the `pip3 --version` command. If you're using Python 2, you'll use the `pip --version` command instead.

```
C:\Users\codecademy>pip3 --version
pip 21.2.4 from C:\Program Files\...\lib\site-packages\pip (python 3.10)
If you instead see:
```



```
'pip3' is not recognized as an internal or external command, operable program or batch file.
```

you may need to update or reinstall Python. [This article about Installing Python 3 and Python Packages can help.](#)

3. Next, let's install **pipenv** using the **pip3 install --user pipenv** command:

```
C:\Users\codecademy>pip3 install --user pipenv
```

You may see some warnings about certain directories not being on **PATH**. This means, if we try the **pipenv** command, it might not work!

```
WARNING: The scripts pipenv-resolver.exe and pipenv.exe are installed in
'C:\Users\codecademy\...\local-packages\Python310\Scripts' which is not on
PATH.                                     Consider adding this
directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-
location.
```

```
...
```

```
C:\Users\codecademy>pipenv
```

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

Let's fix that!

4. To add the directory containing **pipenv** to your **PATH**, return to the Windows Search Bar and type "environment".
 - Choose the option "Edit the system environment variables"
 - Click "Environment Variables..."
 - In the user variables click on the "Path" variable and then click "Edit..."
 - Click "New" and paste the directory path for **pipenv**
 - Click "OK" three times to accept the changes and leave the Settings Menu
5. Close and reopen the Command Prompt to allow the changes made to the environment variables to take effect.

Now, typing **pipenv --version** should work!

```
C:\Users\codecademy>pipenv --version
pipenv, version 2022.1.8
```

Installing **pipenv** on Chromebooks

This process will be very similar to MacOS because they both use a Terminal based on Linux.

1. First let's check that we have **pip** using the **pip3 --version** command. If you're using Python 2, you'll use the **pip --version** command instead.

```
yourusername@penguin:~$ pip3 --version
pip 18.1 from /usr/lib/python3/dist-packages/pip (python 3.7)
```

If you instead see:

```
-bash: pip3: command not found
```

you may need to update or reinstall Python. [This article about programming in Python on a Chromebook can help.](#)

2. Next, let's install **pipenv** using the **pip3 install --user pipenv** command:

```
yourusername@penguin:~$ pip3 install --user pipenv
```

You may see some warnings about certain directories not being on **PATH**. This means, if we try the **pipenv** command, it might not work!

```
The scripts pipenv and pipenv-resolver are installed in
'/home/yourusername/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use
--no-warn-script-location.
```

```
...
```

```
yourusername@penguin:~$ pipenv
-bash: pipenv: command not found
```

Let's fix that!

3. We may need to add the directory **pipenv** is installed in to your **PATH**. We may need to edit our **~/.profile** file using the **vi** editor in our terminal. **If you find yourself getting confused using vi, watch the video above to see someone use vi.**

```
yourusername@penguin:~$ vi ~/.profile
```

This will open a file with some code already in it! Check for the lines:

```
# set PATH so it includes the user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
fi
```

If your file has these lines, you're good to go! Skip to the end of this step where we save and exit the file. If you don't see those lines you will need to add them to your file.

Press the **i** key to enter **INSERT** mode which allows you to type in the file.

At the bottom of the file, add the lines:

```
# set PATH so it includes the user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
fi
```

Then, we need to save and exit the file. To do this, we need to:

- Press the **esc** key to exit **INSERT** mode
- Type **:** which will allow us to enter a **vi** command
- Press the **w** key (to save the file), the **q** key (to exit the file), and **i** to force the command. If this is working correctly, the bottom of the file should look like:

```
if [ -d "$HOME/.local/bin" ] ; then
PATH="$HOME/.local/bin:$PATH"
fi
~
~
:wq!
```

Now, press the **Enter** key.

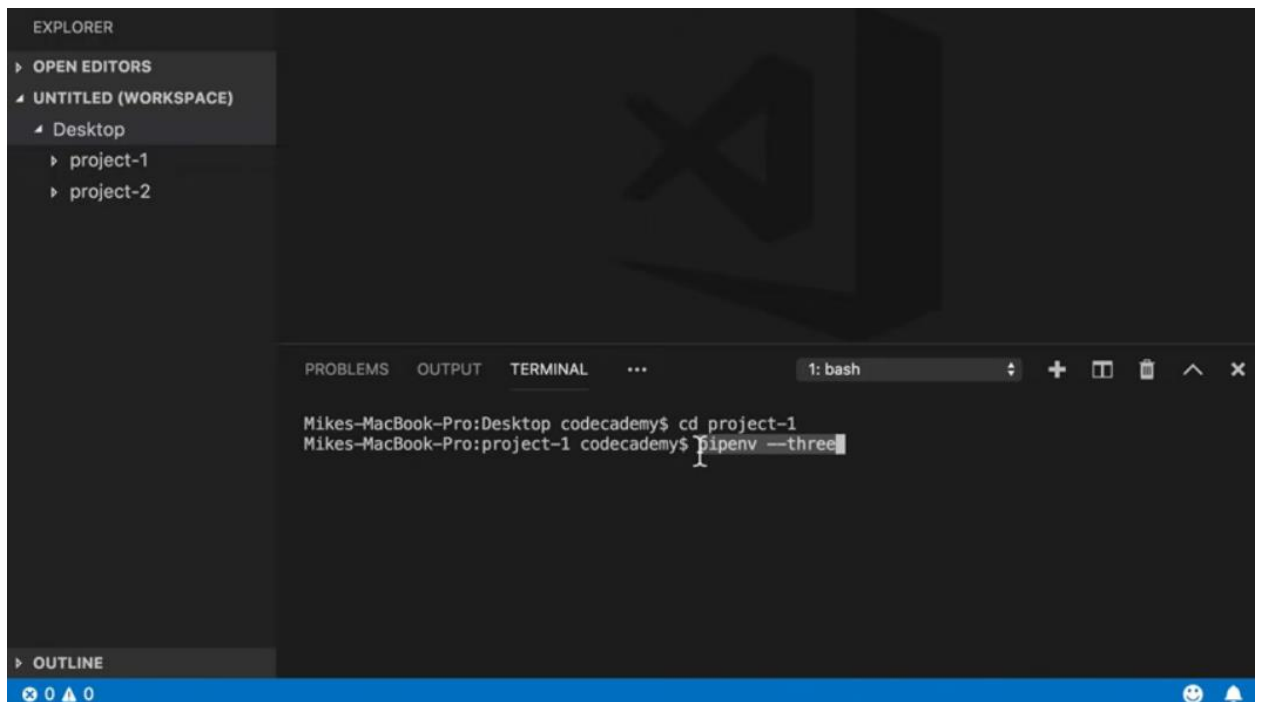
Note: If you don't see the **:** before the **wq!** this means you're typing the letters into the file instead of using a **vi** command. Erase the letters and try pressing the **esc** key to exit **INSERT** mode again.

- Next, we'll use the command `source ~/.profile` to load these environment variables into the current shell.

```
yourusername@penguin:~$ source ~/.profile
Now, typing pipenv --version should work!

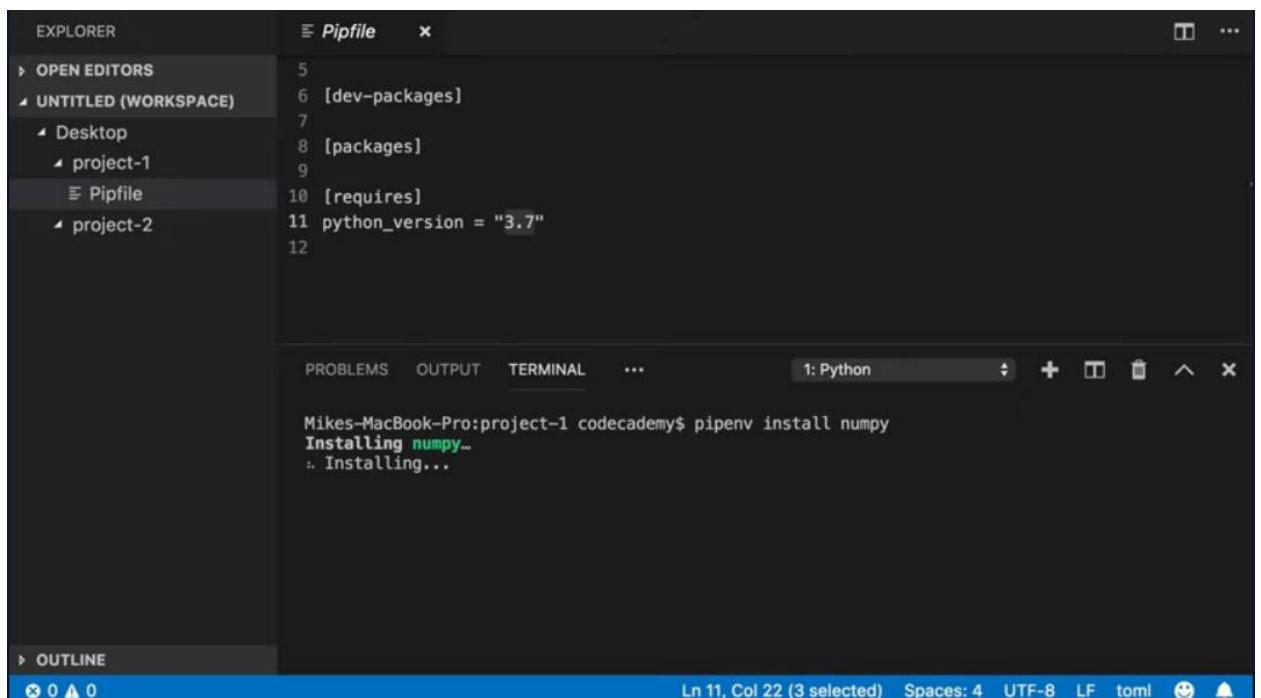
yourusername@penguin:~$ pipenv --version
pipenv, version 2021.5.29
```

This command is going to create a virtual environment for Python 3:



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a workspace with 'Desktop' containing 'project-1' and 'project-2'. The main editor area is dark. At the bottom, the TERMINAL panel is active, showing a bash shell. The prompt is 'Mikes-MacBook-Pro:Desktop codecademy\$'. The user has typed 'cd project-1' and the prompt is now 'Mikes-MacBook-Pro:project-1 codecademy\$'. The user is typing 'pipenv --three'.

Let's start installing some libraries:



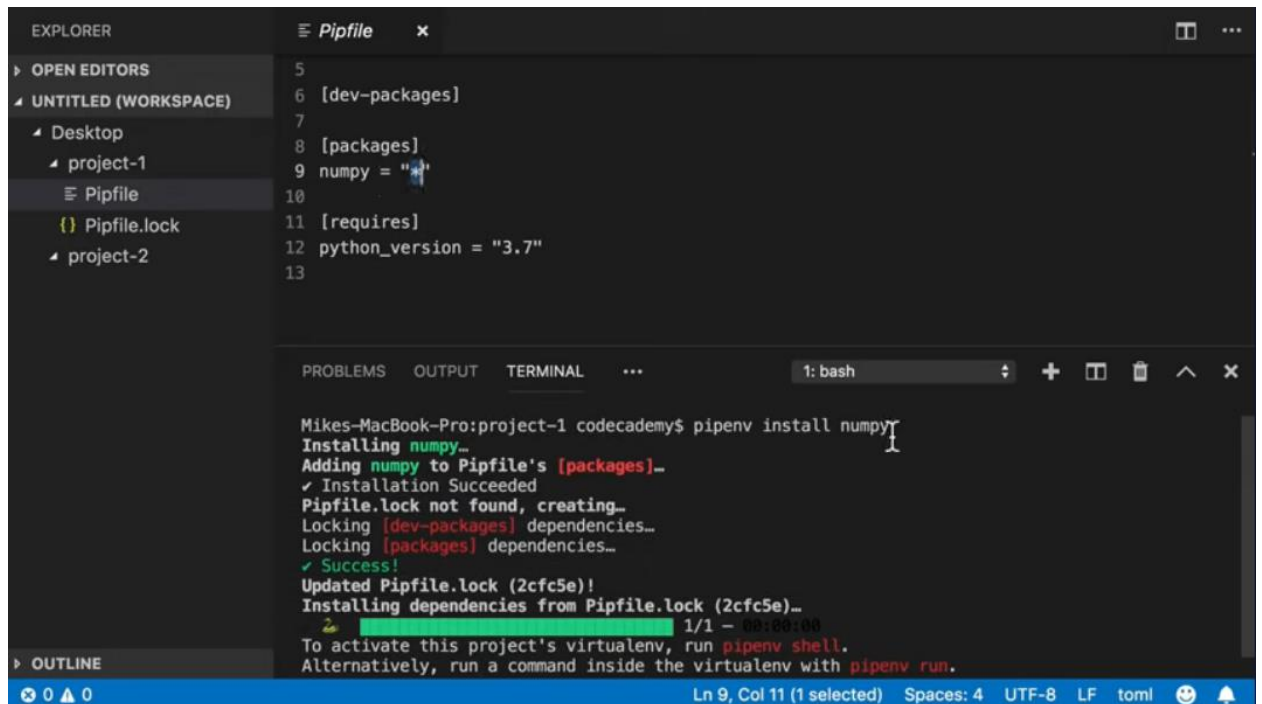
The screenshot shows the Visual Studio Code interface. The Explorer sidebar shows a workspace with 'Desktop' containing 'project-1' and 'project-2'. The main editor area shows a file named 'Pipfile' open. The content of the file is:

```
5
6 [dev-packages]
7
8 [packages]
9
10 [requires]
11 python_version = "3.7"
12
```

At the bottom, the TERMINAL panel is active, showing a Python shell. The prompt is 'Mikes-MacBook-Pro:project-1 codecademy\$'. The user has typed 'pipenv install numpy'. The output shows 'Installing numpy...' and 'Installing...'.

The Pipfile gets updated with this information:

numpy = '*'



The screenshot shows a VS Code editor with a file explorer on the left. The 'Pipfile' is open in the editor, showing the following content:

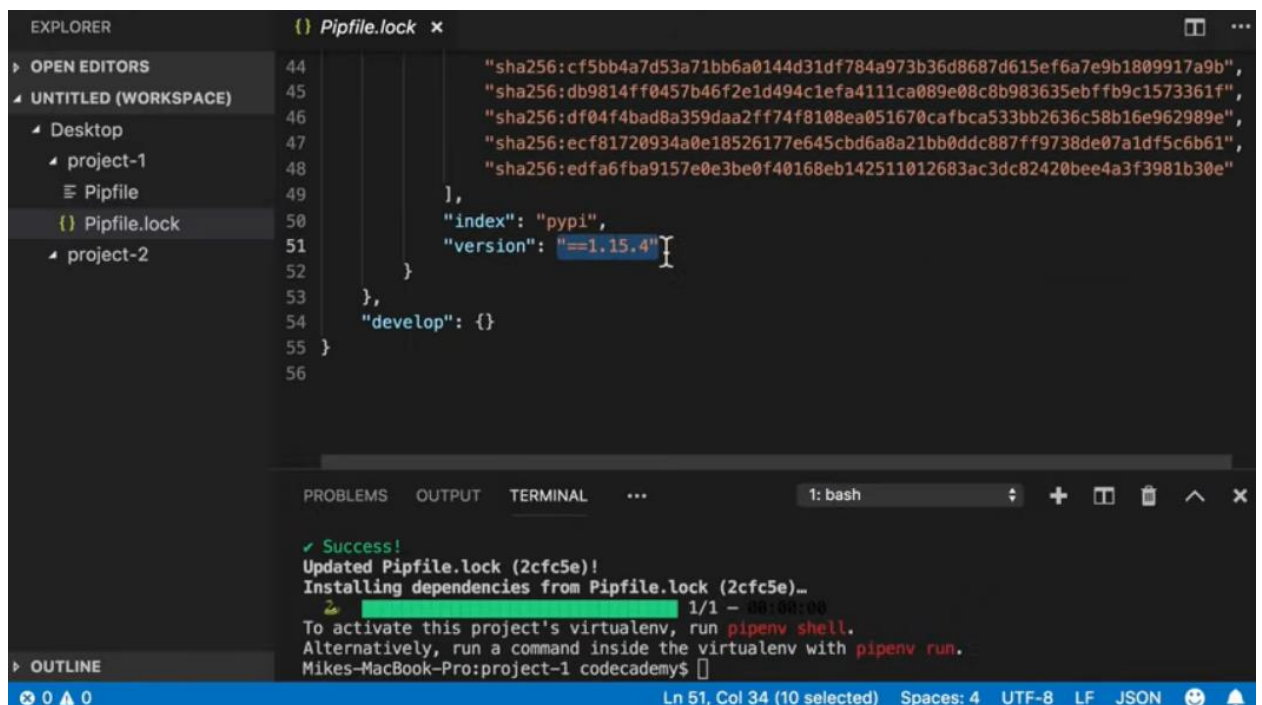
```
5
6 [dev-packages]
7
8 [packages]
9 numpy = "*"
10
11 [requires]
12 python_version = "3.7"
13
```

The terminal at the bottom shows the command `pipenv install numpy` being executed. The output is as follows:

```
Mikes-MacBook-Pro:project-1 codecademy$ pipenv install numpy
Installing numpy...
Adding numpy to Pipfile's [packages]...
✓ Installation Succeeded
Pipfile.lock not found, creating...
Locking [dev-packages] dependencies...
Locking [packages] dependencies...
✓ Success!
Updated Pipfile.lock (2cfc5e)!
Installing dependencies from Pipfile.lock (2cfc5e)...
1/1 - 00:00:00
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
```

That means that this virtual environment supports any version of numpy.

Another file that we see that was created is Pipfile.lock. This specifies all the specific versions of all the different libraries.



The screenshot shows a VS Code editor with the 'Pipfile.lock' file open. The file contains the following JSON content:

```
44
45
46
47
48
49 },
50 "index": "pypi",
51 "version": "==1.15.4"
52 }
53 },
54 "develop": {}
55 }
56
```

The terminal at the bottom shows the command `pipenv install numpy` being executed. The output is as follows:

```
✓ Success!
Updated Pipfile.lock (2cfc5e)!
Installing dependencies from Pipfile.lock (2cfc5e)...
1/1 - 00:00:00
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Mikes-MacBook-Pro:project-1 codecademy$
```

Another thing that we can do is to run the command `pipenv shell`:

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a workspace with two projects, 'project-1' and 'project-2', each containing a 'Pipfile' and a 'Pipfile.lock'. The main editor area displays the 'Pipfile' for 'project-2' with the following content:

```
1 [[source]]
2 name = "pypi"
3 url = "https://pypi.org/simple"
4 verify_ssl = true
5
6 [dev-packages]
7
8 [packages]
9 numpy = "==1.15.2"
10 requests = "*"
11
12 [requires]
```

Below the editor, the Terminal panel is active, showing the command 'pipenv shell' being executed. The output indicates that a subshell is being launched in the virtual environment. The terminal prompt is now '(project-2) bash-3.2\$'.

```
Mikes-MacBook-Pro:project-2 codecademy$ pipenv shell
Launching subshell in virtual environment...
bash-3.2$ . /Users/codecademy/.local/share/virtualenvs/project-2-TASKUss3/bin/activate
(project-2) bash-3.2$
```

Type `exit()` to get out of the shell

We can type in project-1 and project-2:

Import requests or import numpy

`print(requests.__version__)` or `print(numpy.__version__)`

To test the functionality.