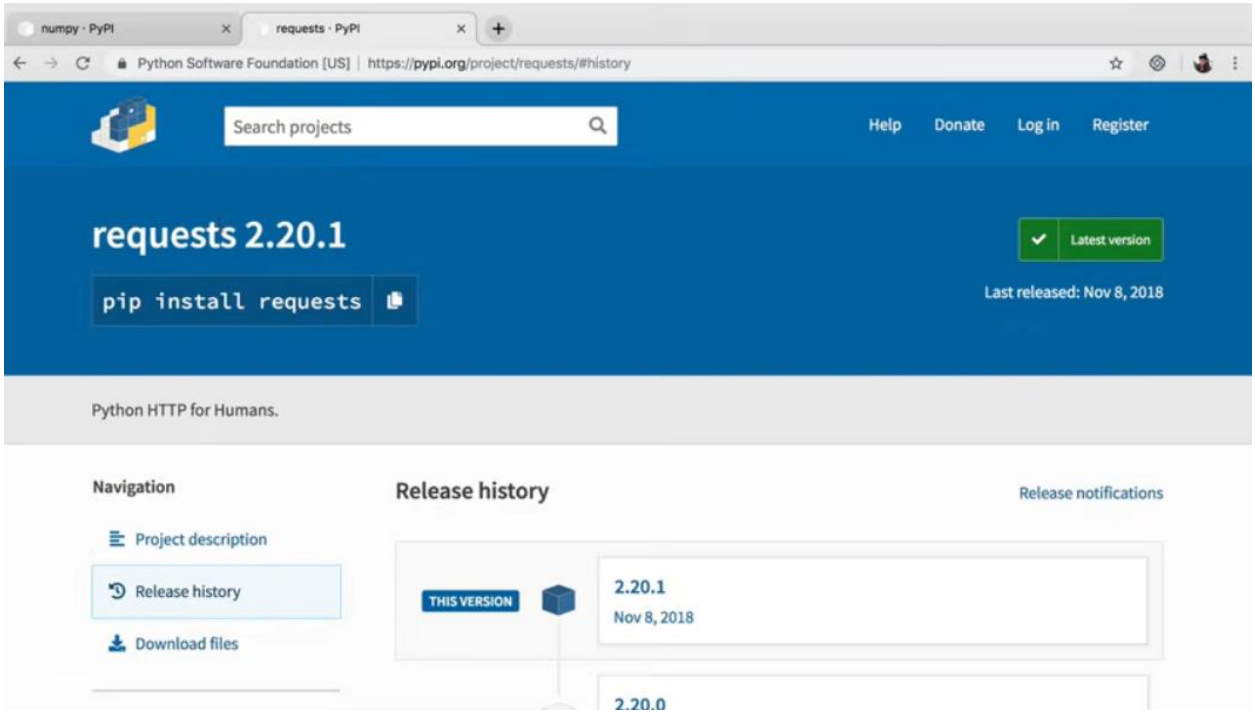
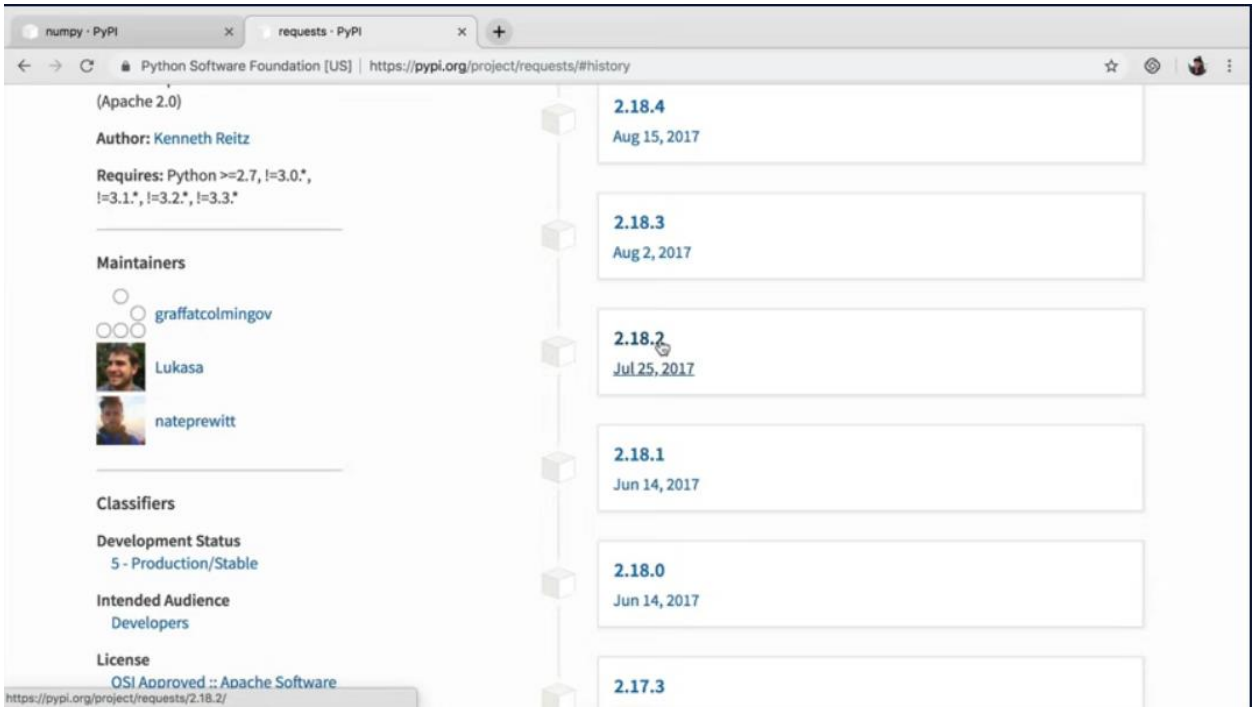
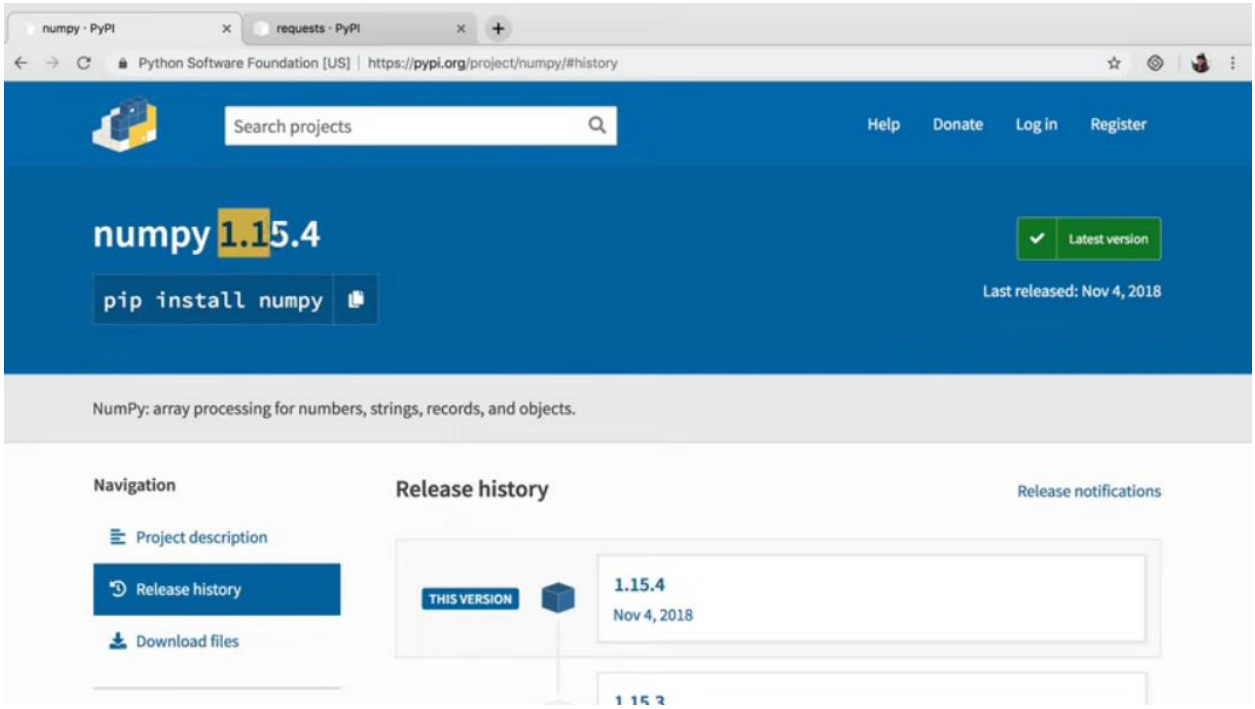
Learn Python: pipenv

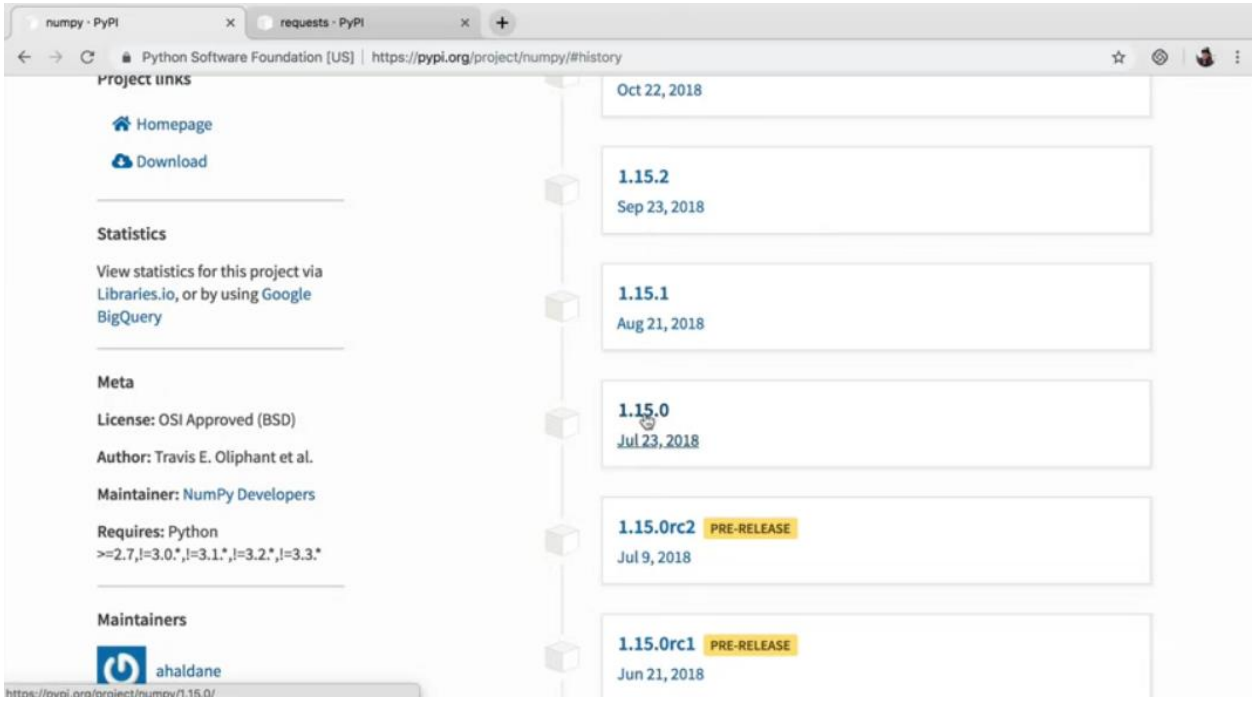
In this article, you'll learn how to install pipenv on Windows, MacOS, and Chromebook computers.

Installing `pipenv` on MacOS and Windows


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
pip install requests
Latest version
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
https://pypi.org/project/requests/2.18.2/
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





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 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 "$HOME/.local/bin" ] ; then
export PATH="$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.

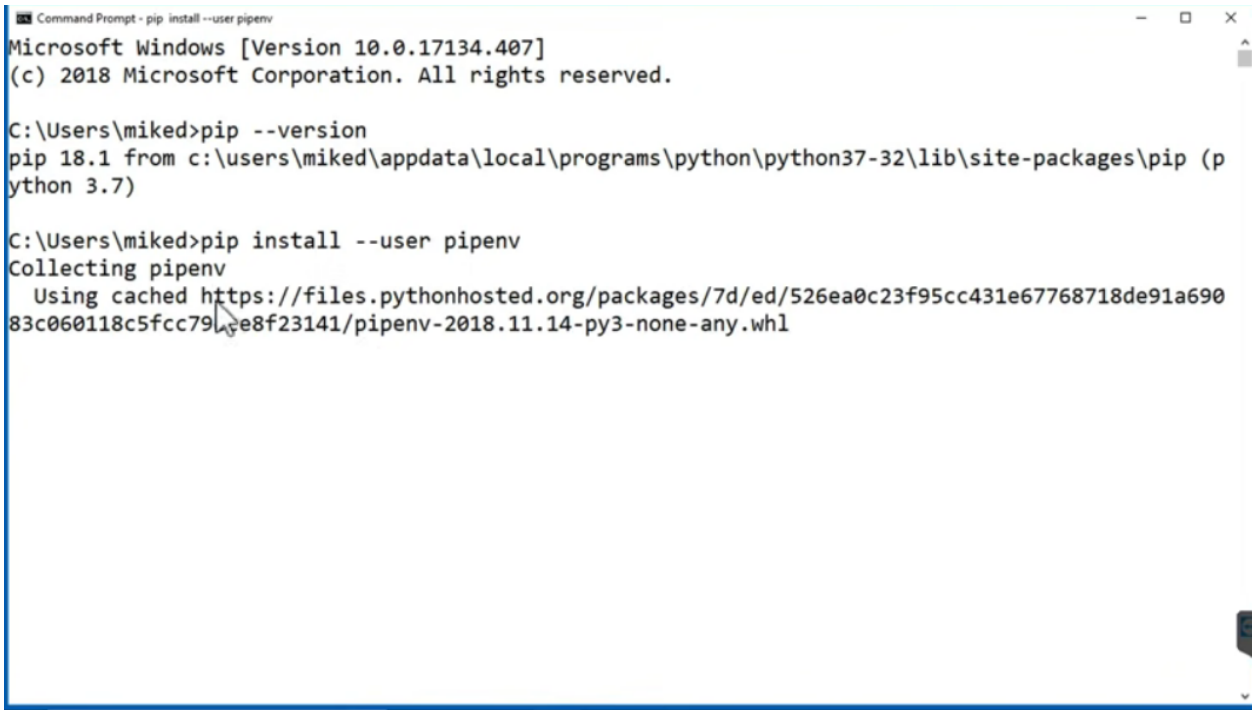
4. Next, we'll use the command `source ~/.bash_profile` to load these environment variables into the current shell.

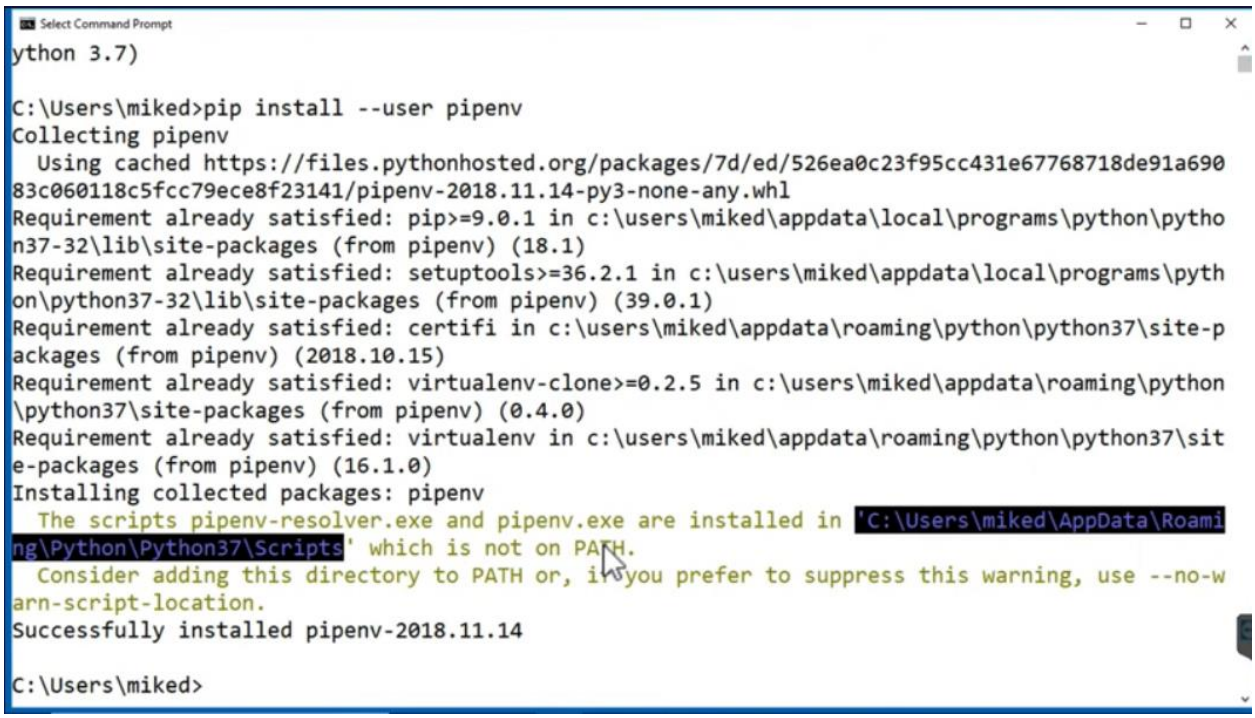
```
My-Mac:~ codecademy$ source ~/.bash_profile
```

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

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

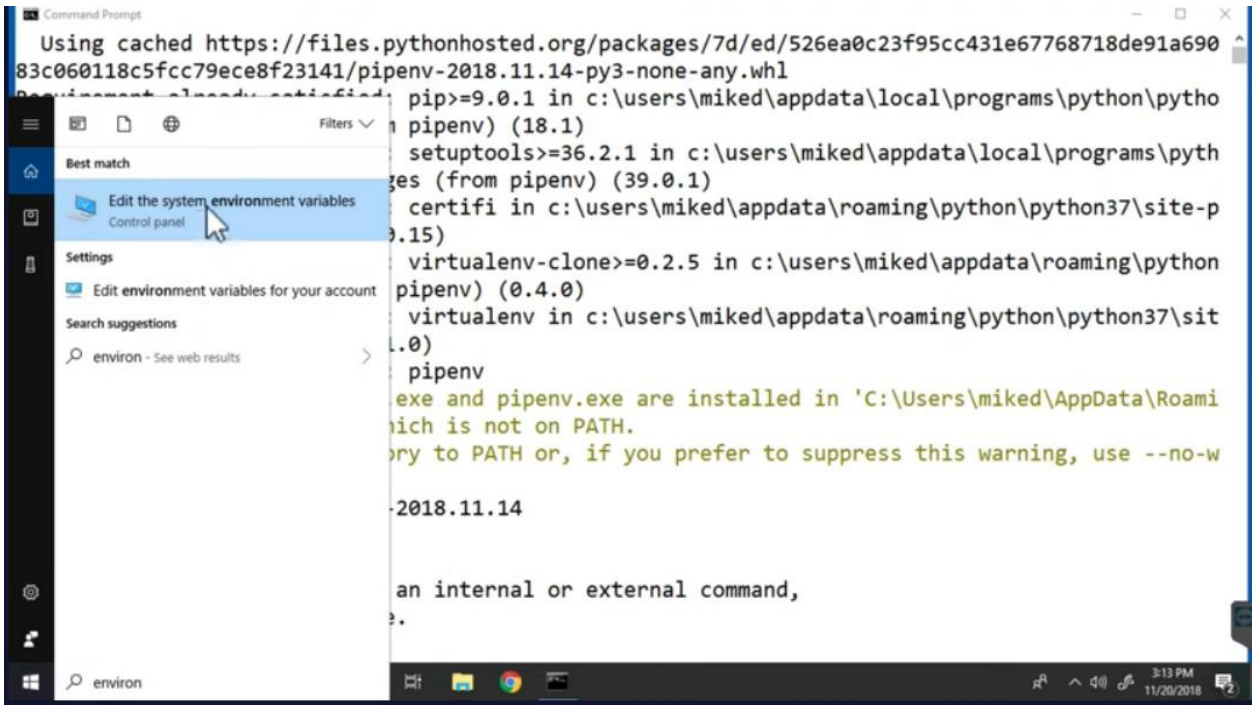
Installing `pipenv` on Windows

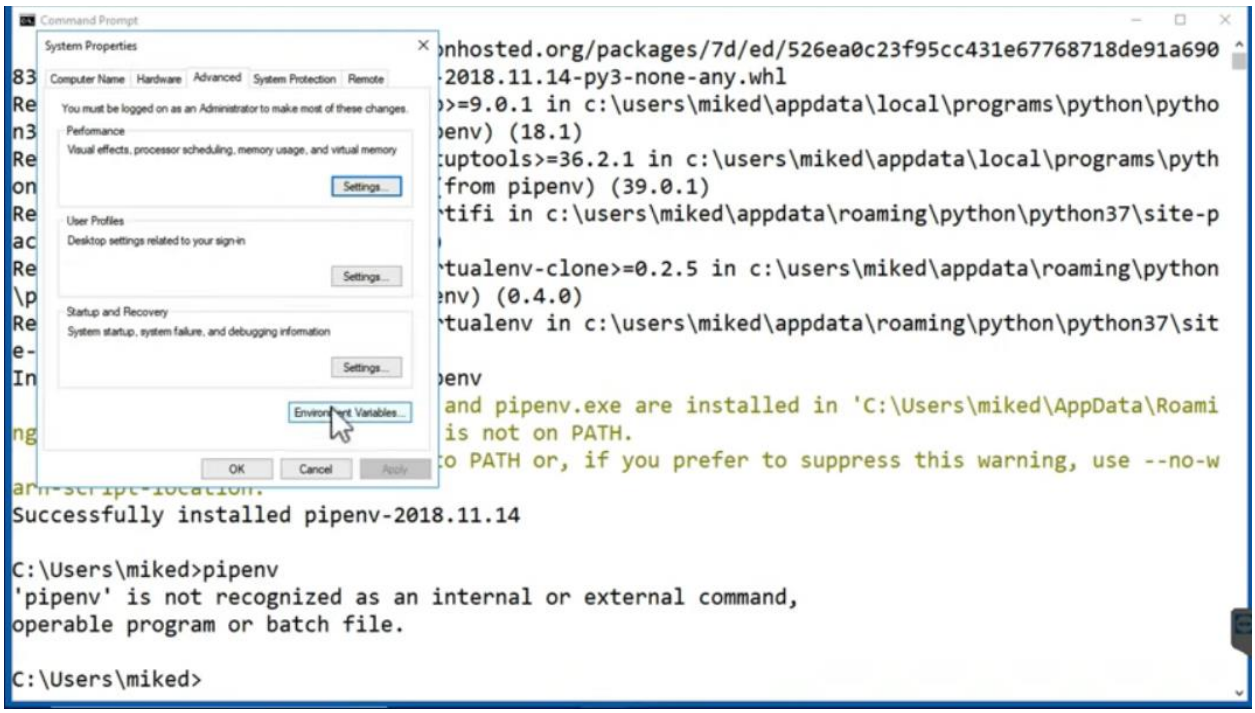


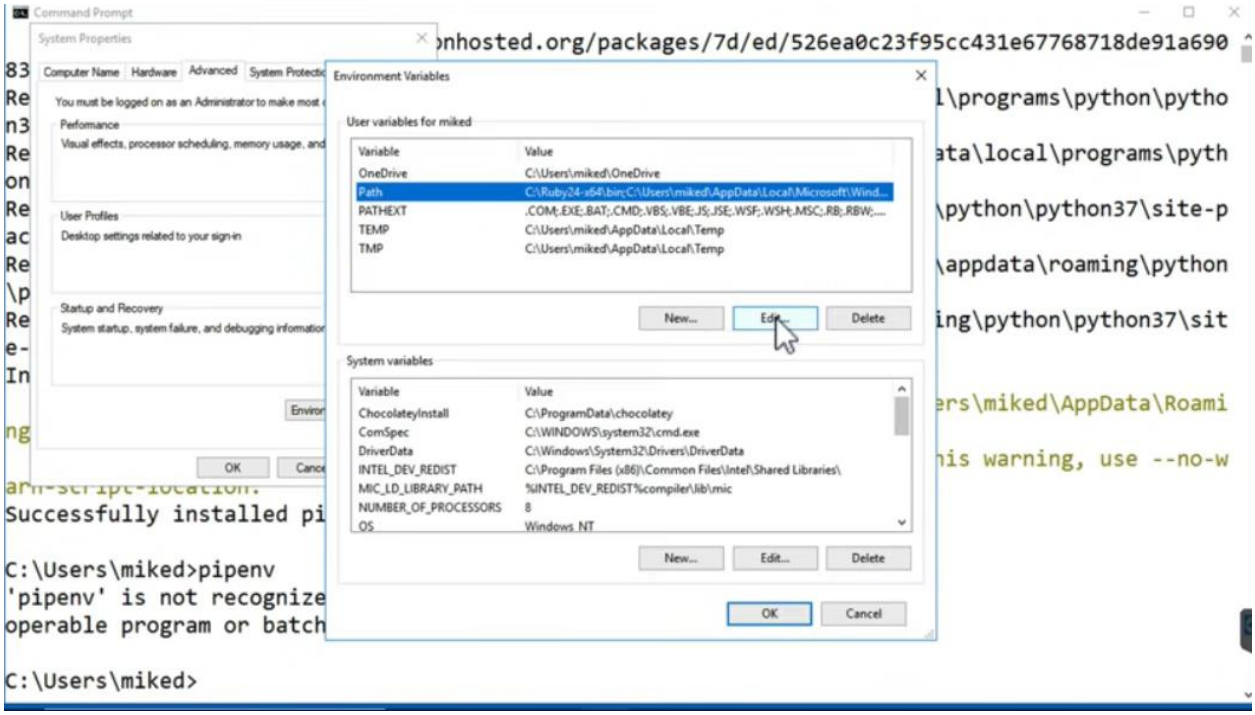


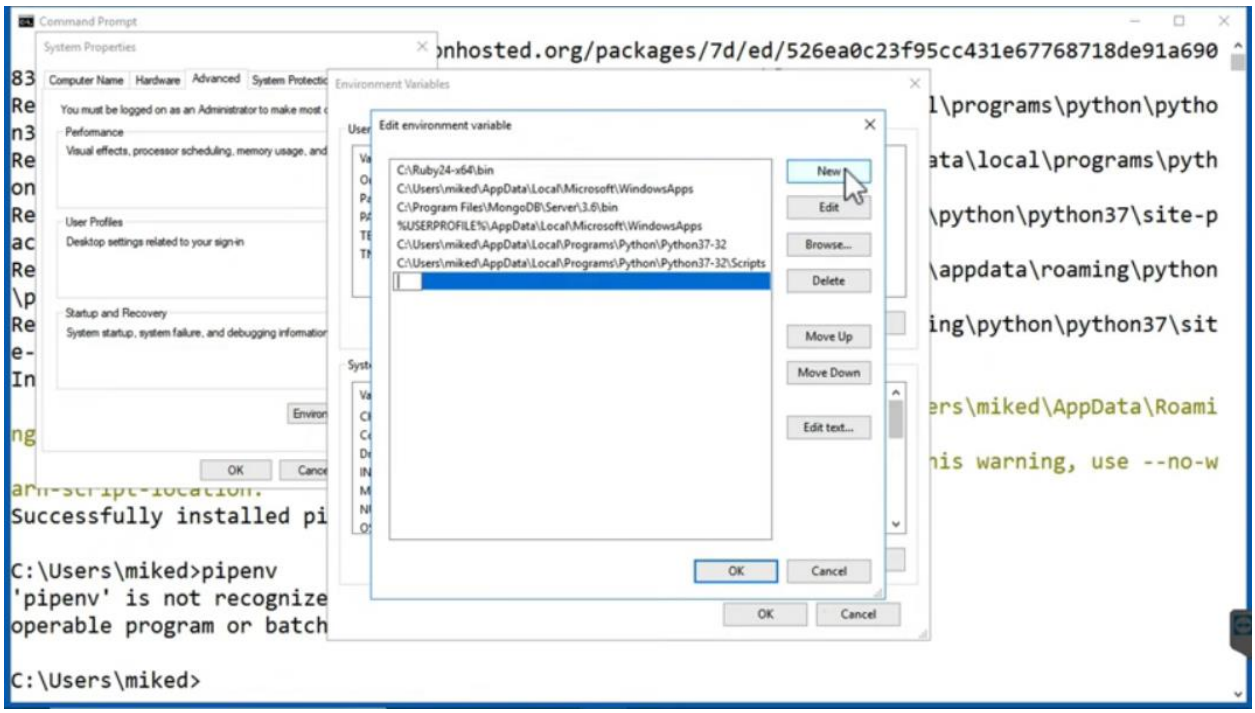
We see a warning: The scripts pipen-resolver.exe and pipenv.exe are installed in ….. which is not on PATH.

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




Command Prompt
System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply
nhosted.org/packages/7d/ed/526ea0c23f95cc431e67768718de91a690
2018.11.14-py3-none-any.whl
>=9.0.1 in c:\users\miked\appdata\local\programs\python\pytho
penv) (18.1)
uptools>=36.2.1 in c:\users\miked\appdata\local\programs\pyth
from pipenv) (39.0.1)
tifi in c:\users\miked\appdata\roaming\python\python37\site-p
tualenv-clone>=0.2.5 in c:\users\miked\appdata\roaming\python
nv) (0.4.0)
tualenv in c:\users\miked\appdata\roaming\python\python37\sit
penv
and pipenv.exe are installed in 'C:\Users\miked\AppData\Roami
is not on PATH.
o PATH or, if you prefer to suppress this warning, use --no-w
Successfully installed pipenv-2018.11.14
C:\Users\miked>pipenv
'pipenv' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\miked>


Command Prompt
System Properties
Environment Variables
User variables for miked
Variable
Value
OneDrive
C:\Users\miked\OneDrive
Path
C:\Ruby24-x64\bin;C:\Users\miked\AppData\Local\Microsoft\Wind...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.RB;.RBW;...
TEMP
C:\Users\miked\AppData\Local\Temp
TMP
C:\Users\miked\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
ChocolateyInstall
C:\ProgramData\chocolatey
ComSpec
C:\WINDOWS\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
INTEL_DEV_REDIST
C:\Program Files (x86)\Common Files\Intel\Shared Libraries\
MIC_LD_LIBRARY_PATH
%INTEL_DEV_REDIST%compiler\lib\mic
NUMBER_OF_PROCESSORS
8
OS
Windows_NT
New...
Edit...
Delete
OK
Cancel
nhosted.org/packages/7d/ed/526ea0c23f95cc431e67768718de91a690
l\programs\python\pytho
ata\local\programs\pyth
\python\python37\site-p
\appdata\roaming\python
ing\python\python37\sit
ers\miked\AppData\Roami
his warning, use --no-w
Successfully installed pi
C:\Users\miked>pipenv
'pipenv' is not recognize
operable program or batch
C:\Users\miked>



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

Installing `pipenv` on Windows

1. Open the Command Promt 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 `!` 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.

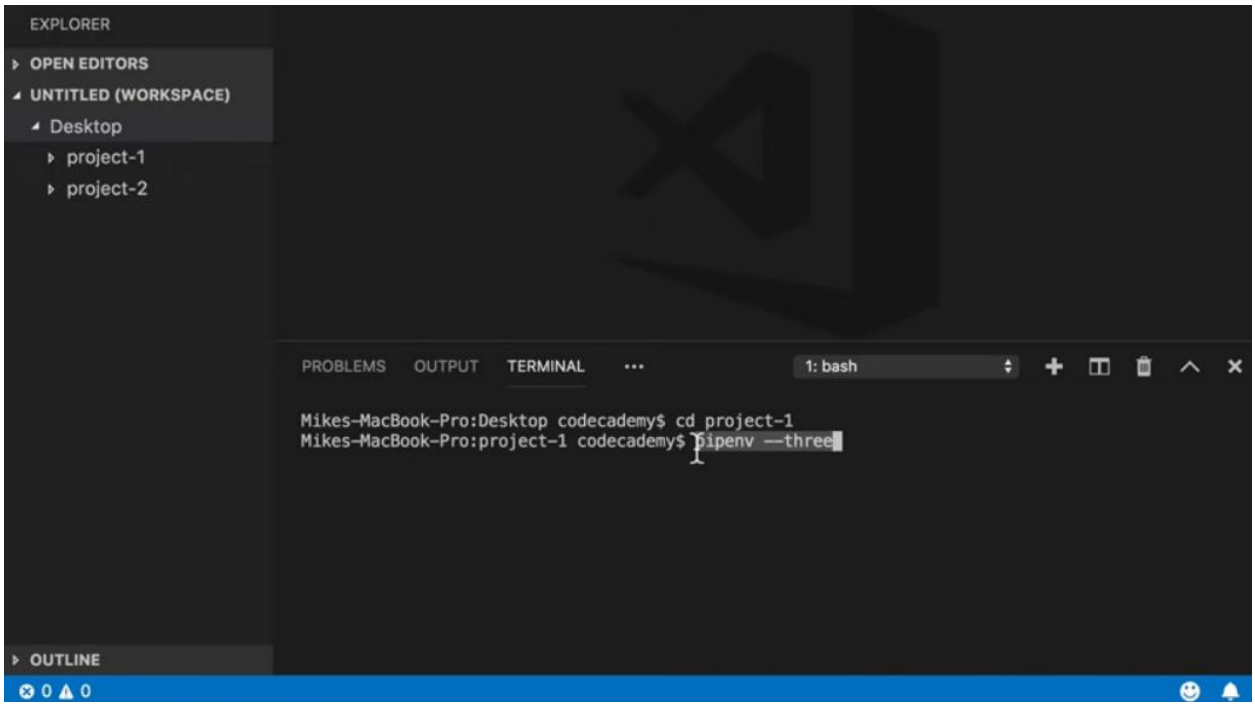
4. Next, we'll use the command `source ~/.profile` to load these environment variables into the current shell.

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

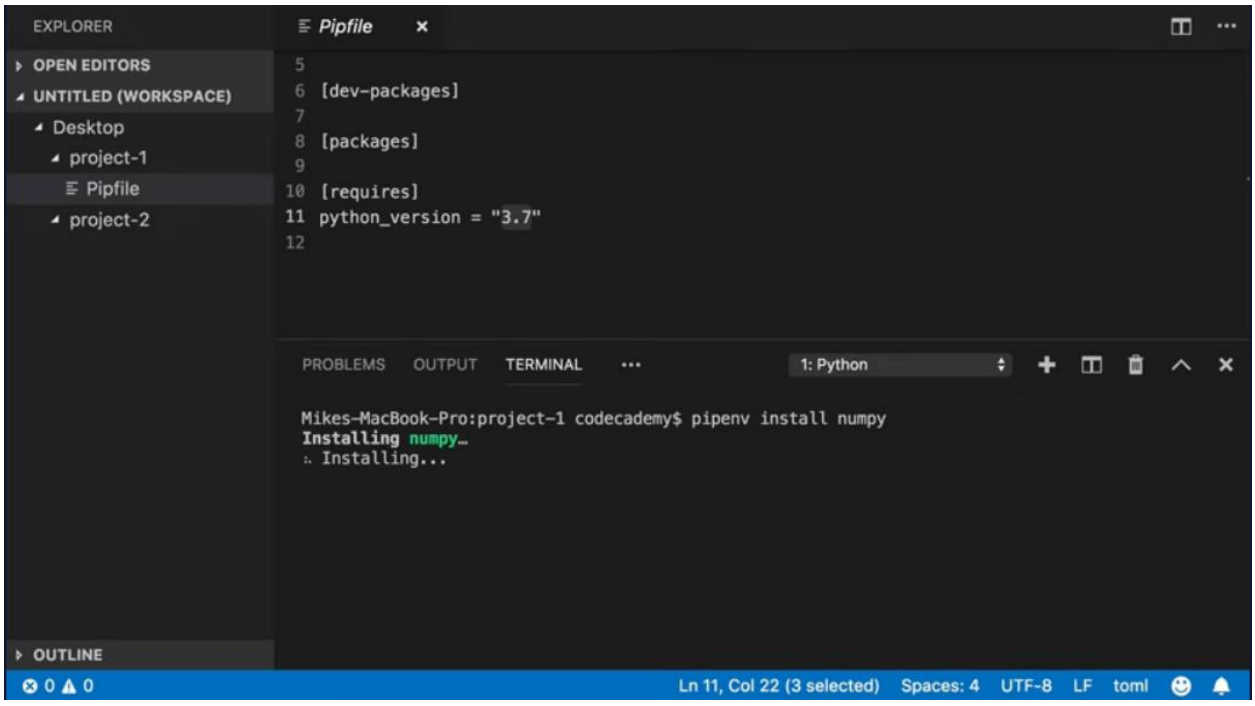
Now, typing `pipenv --version` should work!

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

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

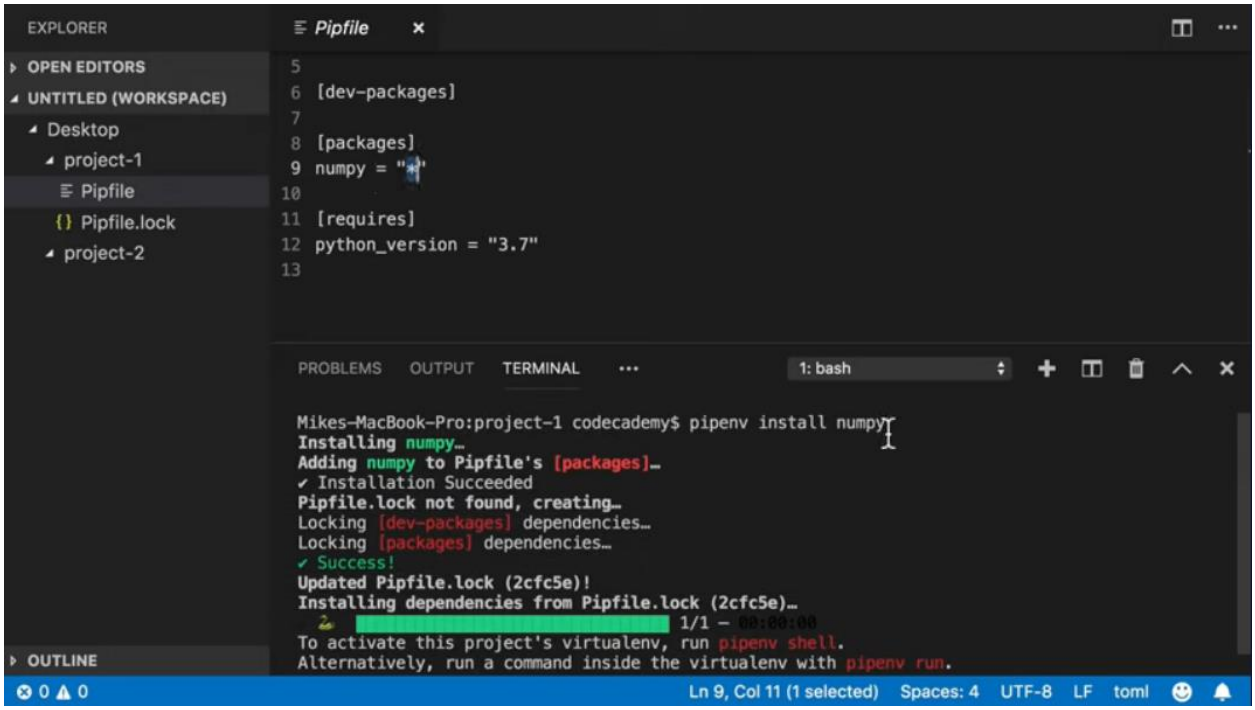


Let's start installing some libraries:



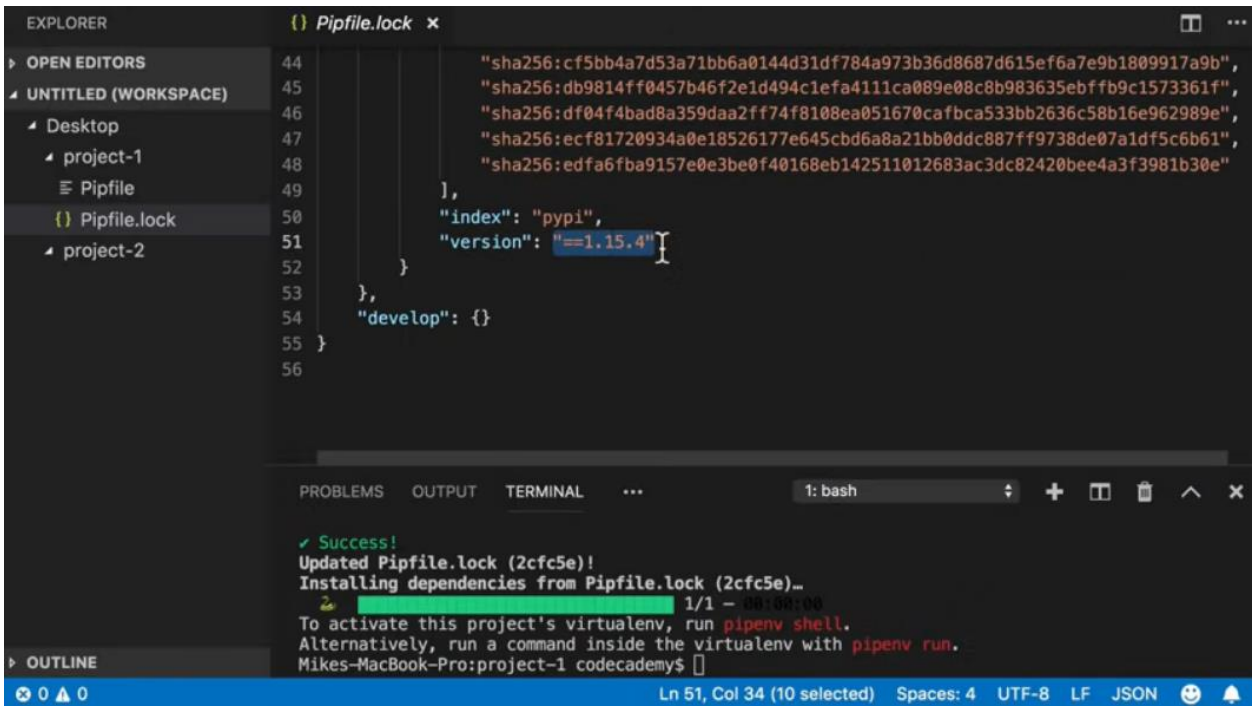
The Pipfile gets updated with this information:

numpy = ‘*’

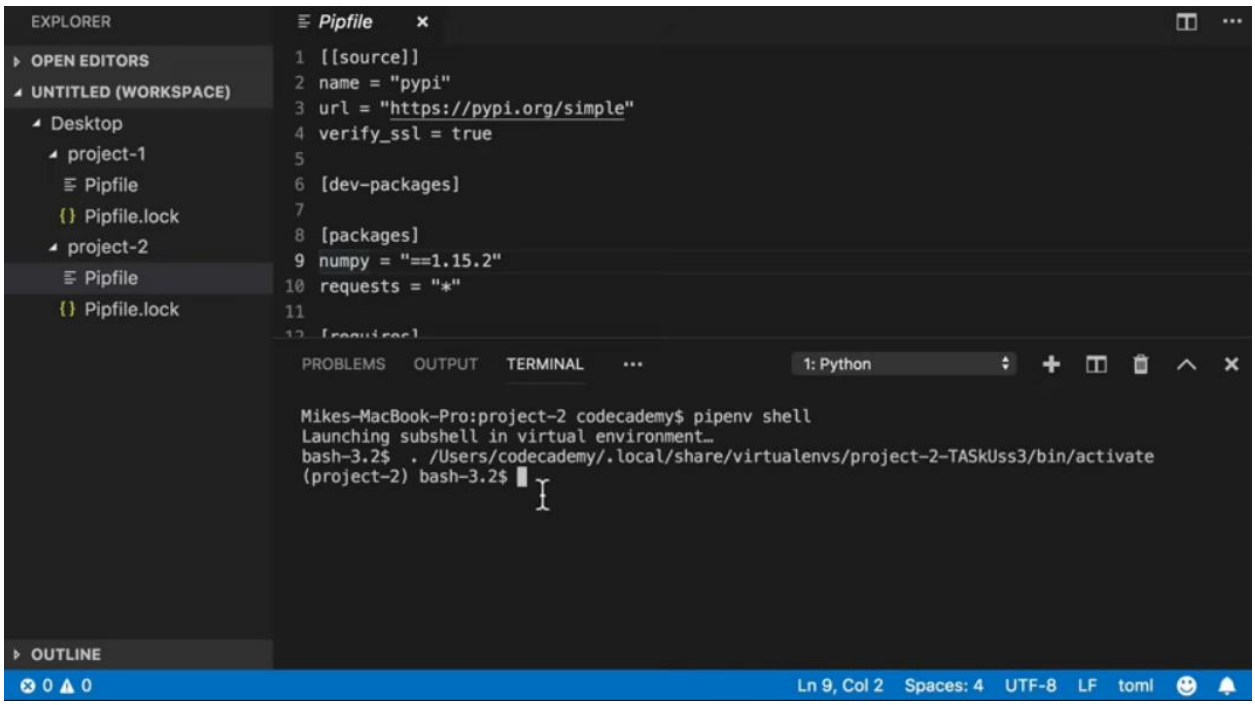


That means that this virtual environment supports any version of numpy.

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



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



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.