



Krishna Regmi

Follow

Apr 19, 2019 · 3 min read · Listen



Save



Pipenv vs virtualenv vs conda environment

Python has three popular ways of creating virtual environment at the moment. In this post I want to talk about how to use each.



Virtual Environment

Virtualenv was the default way of creating virtual environment for many years. It is still used by many although people are moving to improved pipenv or conda (explained below).

Here is how you create a new virtual environment. We will talk about why it got replaced by pipenv after that. Pip comes pre-installed for most newer versions (Python 2.7.9+ or 3.4+). But if you don't have it installed, you can install it by downloading `get-pip.py` file from <https://bootstrap.pypa.io/>. Then cd into the downloads directory and run the file as follows:

```
cd /path/to/downloadsfolder
python get-pip.py
```





Once you have virtualenv installed, you can `cd` into the directory of your choice in the terminal or command prompt, and then run the following:

```
virtualenv venv
```

This will create a folder called `venv` in the directory. You can name your virtualenv folder anything you'd like. To activate run (works in windows only)

```
venv\Scripts\activate
```

There you go! that's how you create a new environment. `pip install package` to install any package you need.

To make your repo reusable, make sure to create a record of everything that's installed in your new environment, run

```
pip freeze > requirements.txt
```

If you are creating a new virtual environment from a `requirements.txt` file, you can run

```
pip install -f requirements.txt
```

If you open your `requirements` file you will see a different package with its version in each line.

As you can see we are creating a virtual environment and then using `pip` to install packages, and then manually calling `pip freeze` to save what's been installed. What if you didn't have to make this a two part process? What if you could merge `pip` with `virtualenv`?

Enter `pipenv`.

PipEnv

Pipenv was created due to many shortcomings of virtualenv such as it not making a distinction if project dependency and the dependencies of the project dependency, not having mechanism to distinguish dev and production needs etc.

To install pipenv, you need to install `pip` first. Then do

```
pip install pipenv
```

Next, you create a new environment by using

```
pipenv install
```

This will look for a `pipenv` file, if it doesn't exist, it will create a new environment and activate it. As you can already see, the workflow





To install new packages do `pip install package` , and pipenv will automatically add the package to the pipenv file that's called `Pipfile` . You can also install package for just the dev environment by calling

```
pip install <package> --dev
```

I advise anyone to actually use Pipenv instead of virtualenv.

Conda Environment

If you are an engineer, or a scientist or use Numpy/Scipy package in windows environment, you have probably experienced the frustration of having to do a lot of work to install numpy/scipy packages. Anaconda is a distribution of python that makes it super simple to install those packages. Anaconda also has their own virtual environment system called conda. Make sure to have Anaconda installed.

To create an environment call this command:

```
conda create --name environment_name python=3.6
```

You can save all the info necessary to recreate the environment in a file by calling

```
conda env export > environment.yml
```

To recreate the environment you can do the following:

```
conda env create -f environment.yml
```

I hope this was useful! Please leave questions in the comments.

