Feb. 05, 2013

Basics (/basics/)

How to use Python virtualenv

What is Virtualenv?

```
A Virtual Environment, put simply, is an isolated working copy of Python which
allows you to work on a specific project without worry of affecting other projects

It enables multiple side-by-side installations of Python, one for each project.

It doesn't actually install separate copies of Python, but it does provide a
clever way to keep different project environments isolated.
```

Verify if Virtualenv is installed

```
There is a chance that virtualenv is already installed on your system.

Run the following command in your terminal
```

```
virtualenv --version
```

```
If you see a version number (in my case 1.6.1), it's already installed.
>>1.6.1
```

Install Virtualenv

```
There are a number of ways to install virtualenv on your system.
```

```
$ sudo apt-get install python-virtualenv

$ sudo easy_install virtualenv

$ sudo pip install virtualenv
```


Setup and Use Virtualenv

```
Once you have virtualenv installed, just fire up a shell and create your own
environment.
```

First create a directory for your new shiny isolated environment

```
mkdir ~/virtualenvironment
```

To create a folder for your new app that includes a clean copy of Python,
simply run:

```
virtualenv ~/virtualenvironment/my_new_app
```

(add --no-site-packages if you want to isolate your environment from the main site
packages directory)

To begin working with your project, you have to cd into your directory (project)
and activate the virtual environment.

```
cd ~/virtualenvironment/my_new_app/bin
```

Lastly, activate your environment:

```
source activate
```

Notice how the prompt of your shell changed to show the active environment.

That is how you can see that you're in your new environment.

Any packages you install now using pip or easy_install get installed into
my_new_app/lib/python2.7/site-packages.

To exit your virtualenv just type **“deactivate”**.

What did Virtualenv do?

Packages installed here will not affect the global Python installation.

Virtualenv does not create every file needed to get a whole new python environment

It uses links to global environment files instead in order to save disk space end
speed up your virtualenv.

Therefore, there must already have an active python environment installed on your
system.

Install a package in your Virtualenv

If you look at the bin directory in your virtualenv, you’ll see easy_install which
has been modified to put eggs and packages in the virtualenv’s site-packages
directory.

To install an app in your Virtualenv:

```
pip install flask
```

You don't have to use sudo since the files will all be installed in the virtualenv
/lib/python2.7/site-packages directory which was created as your own user account

That's it, I hope that you learned something from this post

For further reading, please see:
http://flask.pocoo.org/docs/installation/#virtualenv (http://flask.pocoo.org/docs/installation/#virtualenv)
http://pypi.python.org/pypi/virtualenv (http://pypi.python.org/pypi/virtualenv)
Getting-started-with-virtualenv-isolated-python-environments/ (http://mitchfournier.com/2010/06/25/getting-started-with-virtualenv-isolated-python-environments/)

Recommended Python Training – DataCamp (https://www.datacamp.com/courses/tech:python?tap_a=5644-dce66f&tap_s=75426-9cf8ad)

For Python training (https://www.datacamp.com/courses/tech:python?tap_a=5644-dce66f&tap_s=75426-9cf8ad), our top recommendation is DataCamp.

Datacamp (https://www.datacamp.com/courses/tech:python?tap_a=5644-dce66f&tap_s=75426-9cf8ad) provides online interactive courses that combine interactive coding challenges with videos from top instructors in the field.

Datacamp has beginner to advanced Python training that programmers of all levels benefit from.

Read more about:

- Basics (/basics/)

IfiokJr • 4 years ago

Thanks for this useful post. I've been using virtualenvwrapper since it abstracts away most of the annoyances of virtualenv. And I would recommend it to anyone, especially beginners. However, for provisioning my server with salt it seems that virtualenv is the simpler solution..

3 • Reply • Share ›

Håvard Pedersen • a year ago

I would rather create the virtualenv inside the project folder and exclude it from .gitignore.

1 • Reply • Share ›

OnlineSafetyDepot.com • 4 months ago

Here's a simpler alternative to using virtualenv: https://content.nanobox.io/...

In the Nanobox case, the environments are stored along with the codebase containerized in Docker containers.

• Reply • Share ›

Anirudh Shukla • a year ago

Thanks , very useful :)

• Reply • Share ›

walnutplease . • 4 years ago

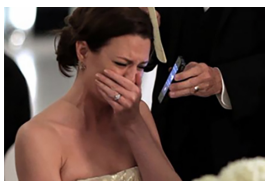
Thanks a lot, this was helpful!

• Reply • Share ›

bra-virus • 4 years ago

great post... got me up and running. ;)

• Reply • Share ›



Disclosure of Material Connection: Some of the links in the post above are "affiliate links." This means if you click on the link and purchase the item, I will receive an affiliate commission. Regardless, PythonForBeginners.com only recommend products or services that we try personally and believe will add value to our readers.