

# Creating a Virtual Environment

## WE'LL COVER THE FOLLOWING ^

- Wrapping Up

Creating a virtual sandbox with the virtualenv package is quite easy. All you need to do is the following:

```
python virtualenv.py FOLDER_NAME
```



Where **FOLDER\_NAME** is the name of the folder that you want your sandbox to go. On my Windows 7 machine, I have **C:\Python34\Scripts** added to my path so I can just call **virtualenv.py** FOLDER\_NAME without the python part. If you don't pass it anything, then you'll get a list of options printed out on your screen. Let's say we create a project called **sandbox**. How do we use it? Well, we need to **activate**. it. Here's how:

On Posix you would do `source bin/activate` while on Windows, you would do `.\path\to\env\Scripts\activate`. on the command line. Let's actually go through these steps. We'll create the sandbox folder on our desktop so you can see an example. Here's what it looks like on my machine:

```
C:\Users\mdriscoll\Desktop>virtualenv sandbox
New python executable in sandbox\Scripts\python.exe
Installing setuptools.....done.
Installing pip.....done.

C:\Users\mdriscoll\Desktop>sandbox\Scripts\activate
(sandbox) C:\Users\mdriscoll\Desktop>
```



You'll note that once your virtual environment is activated, you'll see your prompt change to include a prefix of the folder name that you created, which is **sandbox** in this case. This lets you know that you're using your sandbox.

Now you can use pip to install other packages to your virtual environment. When you're done, you just call the deactivate script to exit the environment.

There are a couple of flags you can pass to virtualenv when creating your virtual playground that you should be aware of. For example, you can use **-system-site-packages** to inherit the packages from your default Python's site packages. If you want to use distribute rather than setuptools, you can pass virtualenv the **-distribute** flag.

virtualenv also provides a way for you to just install libraries but use the system Python itself to run them. According to the documentation, you just create a special script to do it.

There's also a neat (and experimental) flag called **-relocatable** that can be used to make the folder relocatable. However, this does NOT work on Windows at the time of this writing, so I wasn't able to test it out.

Finally, there's an **-extra-search-dir** flag that you can use to keep your virtual environment offline. Basically it allows you to add a directory to the search path for distributions that pip or easy\_install can install from. In this way, you don't need to have access to the internet to install packages.

## Wrapping Up #

At this point, you should be able to use virtualenv yourself. There are a couple of other projects worth mentioning at this point. There's Doug Hellman's **virtualenvwrapper** library that makes it even easier to create, delete and manage virtual environments and then there's **zc.buildout** which is probably the closest thing to virtualenv that could be called a competitor. I recommend checking them both out as they might help you in your programming adventures.