Test your setup

Now let's test our installation and get familiar with creating & using virtual environments:

- Mac OS X/Linux:

```
$ mkvirtualenv TestEnv
Installing
distribute.....................................
..............................................
..............................................
...........................done.
virtualenvwrapper.user_scripts Creating /Users/lynnroot/Envs/TestEnv/bin/predeactivate
virtualenvwrapper.user_scripts Creating /Users/lynnroot/Envs/TestEnv/bin/postdeactivate
virtualenvwrapper.user_scripts Creating /Users/lynnroot/Envs/TestEnv/bin/preactivate
virtualenvwrapper.user_scripts Creating /Users/lynnroot/Envs/TestEnv/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/lynnroot/Envs/TestEnv/bin/get_env_details
```

- Windows:

```
# Within your ProjectFolder
C:\dataviz\Scripts> virtualenv.py TestEnv
Installing
distribute.....................................
..............................................
..............................................
...........................done.
virtualenvwrapper.user_scripts Creating /Users/lynnroot/Envs/TestEnv/bin/predeactivate
virtualenvwrapper.user_scripts Creating /Users/lynnroot/Envs/TestEnv/bin/postdeactivate
virtualenvwrapper.user_scripts Creating /Users/lynnroot/Envs/TestEnv/bin/preactivate
virtualenvwrapper.user_scripts Creating /Users/lynnroot/Envs/TestEnv/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/lynnroot/Envs/TestEnv/bin/get_env_details
```

Now that you made a virtual environment called `TestEnv`, you should see `(TestEnv)` before your prompt:

```
(TestEnv) $
```

Let's play around with commands for virtualenv:

- Mac OS X/Linux:

```
# deactivate the TestEnv
(TestEnv) $ deactivate
$
# reactivate the TestEnv
$ workon TestEnv
(TestEnv) $
# install the Django package in your TestEnv environment
(TestEnv) $ pip install django
Downloading/unpacking django
  Downloading Django-1.1.1.tar.gz (5.6Mb): 5.6Mb downloaded
  Running setup.py egg_info for package django
Installing collected packages: django
  Running setup.py install for django
    changing mode of build/scripts-2.6/django-admin.py from 644 to 755
    changing mode of /Users/lynnroot/Envs/TestEnv/bin/django-admin.py to 755
Successfully installed django
(TestEnv) $
```

- Windows:

```
# deactivate the TestEv
(TestEnv) dataviz\Scripts> deactivate.bat
C:\dataviz\Scripts>
C:\dataviz\Scripts> activate.bat
(TestEnv) C:\dataviz\Scripts>
```

```
# install the Django package in your TestEnv environment
(TestEnv) C:\ pip install django
Downloading/unpacking django
  Downloading Django-1.1.1.tar.gz (5.6Mb): 5.6Mb downloaded
  Running setup.py egg_info for package django
Installing collected packages: django
  Running setup.py install for django
    changing mode of build/scripts-2.6/django-admin.py from 644 to 755
    changing mode of /Users/lynnroot/Envs/TestEnv/bin/django-admin.py to 755
Successfully installed django
(TestEnv) C:\dataviz\Scripts>
```

- All operating systems (for Windows, know that instead of the `$` prompt, you will see `C:\` + folder name:

```
# test the installation of Django
(TestEnv) $ python
Python 2.7.2 (default, Jun 20 2012, 16:23:33)
[GCC 4.2.1 Compatible Apple Clang 4.0 (tags/Apple/clang-418.0.60)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> exit()
# deactivate the TestEnv virtual environment
(TestEnv) $ deactivate
$
```

```
# try to import Django again
$ python
Python 2.7.2 (default, Jun 20 2012, 16:23:33)
[GCC 4.2.1 Compatible Apple Clang 4.0 (tags/Apple/clang-418.0.60)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named django
>>> exit()
$
```

```
# reactivate the TestEnv virtual environment
$ workon TestEnv
(TestEnv) $
# try again to import Django
(TestEnv) $ python
Python 2.7.2 (default, Jun 20 2012, 16:23:33)
[GCC 4.2.1 Compatible Apple Clang 4.0 (tags/Apple/clang-418.0.60)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> exit()
(TestEnv) $
```

```
# see what libraries are installed in the TestEnv virtual environment:
(TestEnv) $ pip freeze
django==1.5
(TestEnv) $
```

- Here's a run-down of useful commands for pip, virtualenv & virtualenvwrapper:
  - For Linux + Mac OS:
    - `mkvirtualenv [ENV_NAME]` - creates and activates a fresh virtual environment
    - `workon [ENV_NAME]` - activates an already-created virtual environment
    - `deactivate` - deactivates the virtual environment that is currently active
    - within an activated virtualenv, `pip install [PACKAGE_NAME]` installs a package into the virtualenv
    - within an activated virtualenv, `pip freeze` lists the packages that is installed & accessible within the virtualenv
  - For Windows:
    - `virtualenv.py [ENV_NAME]` - creates and activates a fresh virtual environment within `ProjectFolder`.
    - `ProjectFolder\Scripts\activate.bat` - activates an already-created virtual environment
    - `ProjectFolder\Scripts\deactivate.bat` - deactivates the virtual environment that is currently active
    - within an activated virtualenv, `pip install [PACKAGE_NAME]` installs a package into the virtualenv
    - within an activated virtualenv, `pip freeze` lists the packages that is installed & accessible within the virtualenv

You're good to go with your setup!