LMS E-LEARNING APP

SETUP BREW, PYTHON 3, VIRTUALENV & POSTGRES IN MAC OSX YOSEMITE

By: Bartlomiej Mika

Date: Feb, 14th, 2015

* BREW \*

(1)

Install Xcode and Homebrew

(2) Install the Command Line Tools of Xcode. Open a Terminal and type:

$ xcode-select –install

(3) Next, we need to install Homebrew. In the Terminal, type this command line:

$ ruby -e “$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/master/install>)”

(4) Now, we need to insert the Homebrew directory at the top of the PATH environment variable. In this way, some Homebrew installations will take precedence over stock OS X binaries. Open or create the file ~/.bash\_profile and write:

- - - - - - - - - - - - - - - - - - - -

Export PATH=/usr/local/bin:$PATH

* - - - - - - - - - - - - - - - - - - -

(5) Close your Terminal and open it again to make these changes effective.

* Python 3 \*

1. Install python 3 (latest)

$ brew install python3

1. You can check which version is installed by typing

$ python3 –version

1. Verify latest is installed:

$ python3

* Virtualenv \*

Q: What is it?

A: Python packages installed via the steps above are global in the sense that they are available across all of your projects. That can be convenient at times, but it can also create problems. For example, sometimes one project needs the latest version of Django, while another project needs an older Django version to retain compatibility with a critical third-party extension. This is one of many use cases that virtualenv was designed to solve. On my systems, only a handful of general-purpose Python packages (such as Mercurial and virtualenv) are globally available — every other package is confined to virtual environments.

1. To install:

$ pip install virtualenv

1. Create some directories to store our projects and virtual environments, respectively:

$ mkdir -p ~/Developer ~/Virtualenvs

1. We’ll then open the ~/.bashrc file (which may be created if it doesn’t exist yet)…

$ vi ~/.bashrc

… and add some lines to it:

* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

# pip should only run if there is a virtualenv currently activated

Export PIP\_REQUIRE\_VIRTUALENV=true

# cache pip-installed packages to avoid re-downloading

Export PIP\_DOWNLOAD\_CACHE=$HOME/.pip/cache

* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

1. Let’s re-load our bash environment:

$ source ~/.bash\_profile

* Postgres \*

See file ‘postgres.txt’ and follow it.

* SOURCE \*

1) <https://www.computersnyou.com/2960/setup-django-virtualenv-macosx-mavericks/>

2) <http://stackoverflow.com/questions/20251562/how-to-install-django-for-python-3-3>

3) <http://hackercodex.com/guide/python-development-environment-on-mac-osx/>

4) <http://stackoverflow.com/questions/12410113/migrating-virtualenv-and-github-between-computers>