Lecture 1

John C. Pyun

Spring 2016

1 Downloading pip (For Linux/ Mac OS)

Pip is a package management system used to install and manage software packages written in Python.

Downloading pip is slightly different and a bit complicated for Windows OS, so please come talk to me if you have Windows OS.Once you have pip installed on WindowsOS, every other steps should almost be exactly the same.

To download pip on your computer, open the terminal and execute the following command:

```
# In the shell
$ sudo easy_install pip
```

Now, there is a method for installing pip without using ”sudo”. If you want to try, please use this website: https://forcecarrier.wordpress.com/2013/07/26/installing-pip-virutalenv-in-sudo-free-way/

Pip makes it extremely easy for peolpe to download any python packages. For example, If you want to install django, you would write:

```
# IMPORTANT::::: DON'T DO THIS YET
$ pip install django
```

But we will get to this later. Do NOT download Django quite yet.

To check if your computer properly installed ”pip”, just open the shell and type

```
$ pip
```

If it doesn't have any errors like ”pip Command not found”, then you have successfully downloaded pip! Congratulations!

Now that we have pip, let's install virtualenv.

to install virtualenv, just write the following command.

```
$ pip install virtualenv
```

If that didn't work, try the following command instead:

```
$ sudo pip install virtualenv
```

Notice how the previous command has the word ”sudo” in it. What it does is that it gives you the superuser privileges, allowing you to do any commands you want with your computer. So, sudo gives you full control of your own machine.

Now that we have installed virtualenv, we can now create our own virtual environment space.

Let’s create a virtual environment folder in our desktop.

on the terminal, type in the following order:

```
# In the shell
$ cd
$ cd Desktop
```

Typing the previous command should make your terminal to go into the Desktop directory.

After you are inside Desktop directory, type the following command on your terminal to create a virtual environment folder:

```
# instead of "YOURPROJECTNAME" type anything you want as the name of
    your folder.
$ virtualenv YOURPROJECTNAME
```

Congratulations! You have successfully created a virtual environment folder. Anything you do inside this virtual environment folder will be isolated to this folder. So, if you mess up any python packages, it will only affect this folder rather than your machine.

Now that you have successfully created a virtual environment folder, you now have to ACTIVATE your virtual environment.

This is very important. All of our classwork will be inside our virtual environment. So don’t forget to always activate the virtual environment before doing any coding.

To activate our virtual environment, go inside the virtual environment folder that we have created. To go inside the folder, do the following command:

```
# In the shell
$ cd YOURPROJECTNAME
```

After going inside your virtual environment folder, do the following command to ACTIVATE your virtual environment:

```
# In the shell
$ source bin/activate
```

Congratulations! You have successfully activated your virtual environment folder.

Now, let us download the django package inside our virtual environment.

```
# In the shell
$ pip install django
```

This should successfully download django in our virtual environment.

To check if it worked, type the following command and see if django is installed

```
# In the shell
$ pip freeze
```

if the following shows up, that means you successfully installed django:

Django==1.9.1

wheel==0.24.0

Now, let's create our first django project! Type the following command:

```
# In the shell
$ django-admin startproject helloworld
```

this should create a django project named "helloworld".

now, go inside this "helloworld" directory by typing the following command:

```
# In the shell
$ cd helloworld
```

now, let's start our django server by typing the following command:

```
# In the shell
$ python manage.py runserver
```

This should activate your django server. To check out your django website, go to the following url on your web browser

http://127.0.0.1:8000/

Congratulations! You have your own functioning webapp!

If you are stuck, please e-mail me at jcp@andrew.cmu.edu