Instructions for Installing Python

# Prerequisites:

This guide will help you to install python 3.6, openenv, and pip. You will need all of these materials in conjunction with each other to successfully complete installation and manage the test environment.

**Please ensure the following:**

Be on your administrator account for a Windows Install;

Administrator for Mac;

Normal user on Linux should be okay, *however* if you are getting usage errors, you **need** to prefix your commands with sudo. The steps in this guide will **not** break your computer, but be wary of potential internet hotfixes.

# Side Notes:

This guide was successful on Windows 10 and Linux (Ubuntu 16.04). *No testing completed on Mac*

# Let’s Start!

Linux installation is noted here. Mac is after, and Windows is at the end.

## Linux Installation

First, verify that you have python 3.6 installed.

$ python3.6 –V

If returns an error, then we must install python using the following commands:

$ sudo add-apt-repository ppa:jonathonf/python-3.6

$ sudo apt-get update

$ sudo apt-get install python3.6

Say yes to all the prompts that occur while running these lines. You **won’t** break your computer.

Okay! You have python3.6. Let’s get pip3.6. Type in the following command into your terminal:

$ curl https://bootstrap.pypa.io/get-pip.py | sudo python3.6

If you don’t have curl installed, install it then run the above.

Then, your pip directory should point to the python3.6 target **this is important**, since we will be installing flask, some APIs, etc. etc. that **must be able to be located by the 3.6 interpreter**. Otherwise, the test harness won’t run.

Install virtualenv.

$ pip3.6 install virtualenv

Although using pip instead of pip3.6 should work as well.

Alright! Create a virtual environment for the test-harness:

$ virtualenv 2017

Now this is the fun stuff. In your package, you have the test harness. Do the following:

Change the directory to 2017 and startup the environment.

$ cd 2017

$ source bin/activate

Install the required python libraries (**important: these must be included where python3.6 interpreter will be able to find them and run them. Otherwise, when running the harness, you will get a syntax error.)**

(2017) $ pip3.6 install -r requirements.txt

Install the test harness:

(2017) $ mkdir templates  
(2017) $ mkdir static

Copy the attached files as follows:

index.html -> 2017/templates  
5x5.png -> 2017/static  
test\_harness.py -> 2017/  
naiive-client.py -> 2017/  
5x5.csv -> 2017/

Run the test harness:

Open a second command line window and change to the 2017 directory (the one made with virtualenv)

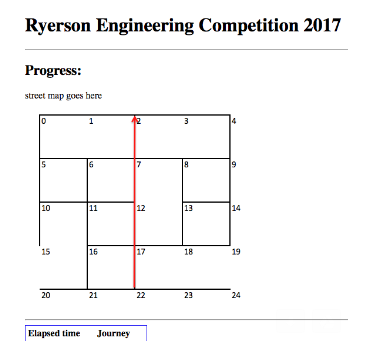
$ cd 2017  
$ source bin/activate

In the first window, start the test-harness:

(2017)$ python3.6 test\_harness.py 5002 5x5.csv

In your browser, go to <http://localhost:5002>

If everything is working, you should see this in your browser.



In the second window, start your client.py program. <name of your program here>

(2017) $ python <>.py

Watch the progress in the browser.

To reset, simply go to the test harness terminal and Ctrl-C twice to kill it, restart it then refresh your browser window.

## Mac Installation

Side note: This assumes you are running an install on Mac OS for Intel Processors.

The first part of this tutorial can also be found on:

<https://wsvincent.com/install-python3-mac/>

First check what version of python you have.

python3 --version

If you see an error message, that is okay. Otherwise, we want to ensure that we are running the *latest* version of python on this computer.

Next, install Xcode (click through all confirmation commands).

$ xcode-select --install

Next, install Homebrew.

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

To confirm that homebrew is installed:

$ brew doctor

Your system is ready to brew.

Now, let us install python.

$ brew install python3

Verify the version that was installed:

$ python3 --version

Python 3.6.3

Let’s install venv now (virtualenv). First, let’s be tidy and create the directory for where we will have all of our venvs. Basically, these repositories will allow you to run python programs in the same machine.

$ mkdir ~/.virtualenvs

$ python3 -m venv ~/.virtualenvs/2017

2017 is the home of the test harness.

Let’s install the test harness.

Here, we need to install pip so that the packages (flask, RESTful, etc.) are installed to the correct directory for the python 3.6 interpreter to run. Otherwise we might run into syntax problems.

Reference: https://stackoverflow.com/questions/34573159/how-do-install-pip3-on-my-mac

Run:

brew postinstall python3

Check the console for errors. If no errors, then we can continue. *If we do run into errors*:

Download the python 3.6.3 installer from the web-link, and choose to get pip3 for your computer. When the installation is completed (no special choices), you should either have commands pip3 or pip3.6. If pip3.6 works for the next step, then continue usage. Otherwise, reinstall and make sure that pip3.6 is installed.

Now that pip is out of the way, we have to install the test harness.

Copy the requirements.txt file to the 2017 directory.

Change the dir to 2017 and startup the environment.

$ cd 2017  
$ source bin/activate

Install the required python libraries:

(2017) $ pip3 install –r requirements.txt

If that leads to a permission error, please append sudo to the beginning, as so:

(2017) $ sudo pip3 install -r requirements.txt

**Make sure you use pip3. Otherwise, you will install to the python 2.7 repo, which is useless to us. We *need* this to go into the python3.6 folder, in order to be able to use these apis.**

Create the following folders inside the 2017 folder:

(2017) $ mkdir templates  
(2017) $ mkdir static

Copy the attached files as follows:

    index.html -> 2017/templates  
    5x5.png -> 2017/static  
    test\_harness.py -> 2017/  
    naiive-client.py -> 2017/  
    5x5.csv -> 2017/

Almost there! Let’s run the test harness.

Open a second terminal window. Change the directory to 2017 and startup the environment.

$ cd 2017

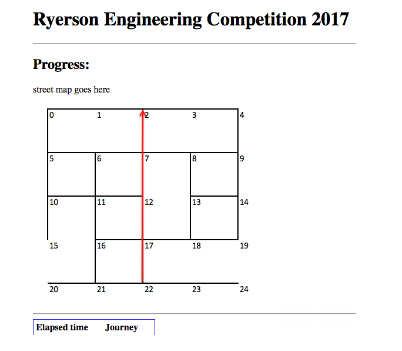
$ source bin/activate

In the first window, start the test-harness:

(2017) python3 test\_harness.py 5002 5x5.csv

In your browser, go to <http://localhost:5002>

If everything works, you should see this in your browser:



Go back to your second window. Start your program <your program name here>

(2017) $ python <>.py

Watch the progress of your program.

To reset, simply go to the test harness wndow and Ctrl C twice to kill it, restart it then refresh your browser window.

## Windows Installation

Ah, the best for last. Adapted from: <http://www.tylerbutler.com/2012/05/how-to-install-python-pip-and-virtualenv-on-windows-with-powershell/>

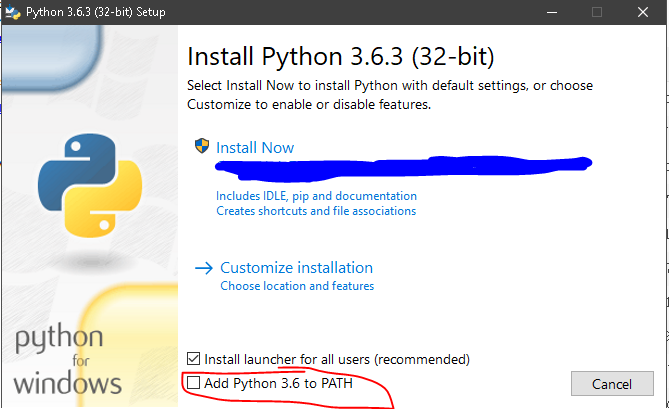
**First: You *need to make sure that powershell can run scripts. If this does not work, you will have to use the*** Set-ExecutionPolicy ***command. Usage can be found here:*** [***https://stackoverflow.com/questions/4037939/powershell-says-execution-of-scripts-is-disabled-on-this-system***](https://stackoverflow.com/questions/4037939/powershell-says-execution-of-scripts-is-disabled-on-this-system)

Revert the setting after this competition; it is a safety measure after all!

Get python from: <https://www.python.org/downloads/release/python-363/>

*Yes, we will be using python 3.6!*

Get the zip files or the installers. Preferably, get the 32 bit version (According to the tutorial, it does not matter). When installing, make sure that:



Make sure to add python as a PATH variable (just tick the box). However, we will eventually have to modify this value since it does not add the scripts folder we need (where pip will install to – more on this)

Click on customize installation. What you want to do here is to check every single checkbox imaginable (getting pip, for example). In the second dialogue, check everything except (install for all users), unless that is important in your specific case.

Wait for the installation to complete.

Okay! Open up power shell as an administrator, and type in the following:

PS C:\> python

Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>>

Press Ctrl-Z, and if you get an error saying the term ‘foo’ is not recognized, then Python is not on your path. There are docs in the documentation as to how to do this, but for time’s sake, uninstall python and **MAKE SURE that that little checkbox is ticked!**

Since pip is now installed, we can plow forwards and install virtualenv.

PS C:\> pip install virtualenv

PS C:\> pip install virtualenvwrapper-powershell

You might get an error that says that setting up an egg failed.

Command "python setup.py egg\_info" failed with error code 1 in C:\Users\sampleUser\AppData\Local\Temp\pip-build-zllta4te\virtualenvwrapper-powershell\

If so, try the answer to this question: <https://stackoverflow.com/questions/35991403/python-pip-install-gives-command-python-setup-py-egg-info-failed-with-error-c>

Additionally, the python documentation is available here: <https://docs.python.org/3/using/windows.html>

**If you run into this problem (no permissions into the folder),** you can attempt using the methods outlined here (way below the beginning of the page). However, this is a difficult problem to solve; it is recommended to switch OS (let a volunteer know!)

Keep going:

Now you need to import the wrapper module in PowerShell, so type

Import-Module virtualenvwrapper

You will probably get one of two errors – or both. The first will be something like this:

PS C:\> Import-Module virtualenvwrapper

Get-Content : Cannot find path 'Function:\TabExpansion' because it does not exist.

Unfortunately that’s a bug in the current released version (12.7.8) of virtualenvwrapper-powershell. It doesn’t actually cause any problems in practice as far as I know.

The other error you might see will say something like this:

Virtualenvwrapper: Virtual environments directory

'C:\Users\sampleuser/.virtualenvs' does not exist. Create it or

set $env:WORKON\_HOME to an existing directory.

Well, at least you know you’re on the right track! Do exactly what the message says: create the missing directory.

mkdir '~\.virtualenvs'

Stick to this location for your environments.

Now try to import the module again. Success! Now you have access to a bunch of virtualenv management commands directly in PowerShell. To see all of them, you can type:

PS C:\> Get-Command \*virtualenv\*

Now that we have virtualenv installed, let’s make a new virtualenv:

New-VirtualEnvironment 2017

After the command completes, you should see a PowerShell prompt that looks like this:

(2017)PS C:\>

The (2017) prepended to your prompt reminds you that you’re currently working within that virtualenv. If you type workon now you should see the available virtualenvs, and if you type workon name\_of\_another\_virtualenv you’ll flip to that environment.

Alright! Now it’s time to install the required packages for the test harness.

Change the dir to 2017 and startup the environment.

$ cd 2017  
$ source bin/activate

Install the required python libraries:

(2017) $ pip3 install –r requirements.txt

If that leads to a permission error, please append sudo to the beginning, as so:

(2017) $ sudo pip3 install -r requirements.txt

**Make sure you use pip3. Otherwise, you will install to the python 2.7 repo, which is useless to us. We *need* this to go into the python3.6 folder, in order to be able to use these apis.**

Create the following folders inside the 2017 folder:

(2017) $ mkdir templates  
(2017) $ mkdir static

Copy the attached files as follows:

    index.html -> 2017/templates  
    5x5.png -> 2017/static  
    test\_harness.py -> 2017/  
    naiive-client.py -> 2017/  
    5x5.csv -> 2017/

Almost there! Let’s run the test harness.

Open a second terminal window. Change the directory to 2017 and startup the environment.

$ cd 2017

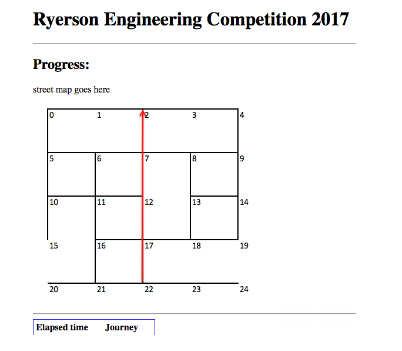
$ source bin/activate

In the first window, start the test-harness:

(2017) python3 test\_harness.py 5002 5x5.csv

In your browser, go to <http://localhost:5002>

If everything works, you should see this in your browser:



Go back to your second window. Start your program <your program name here>

(2017) $ python <>.py

Watch the progress of your program.

To reset, simply go to the test harness window and Ctrl C twice to kill it, restart it then refresh your browser window.