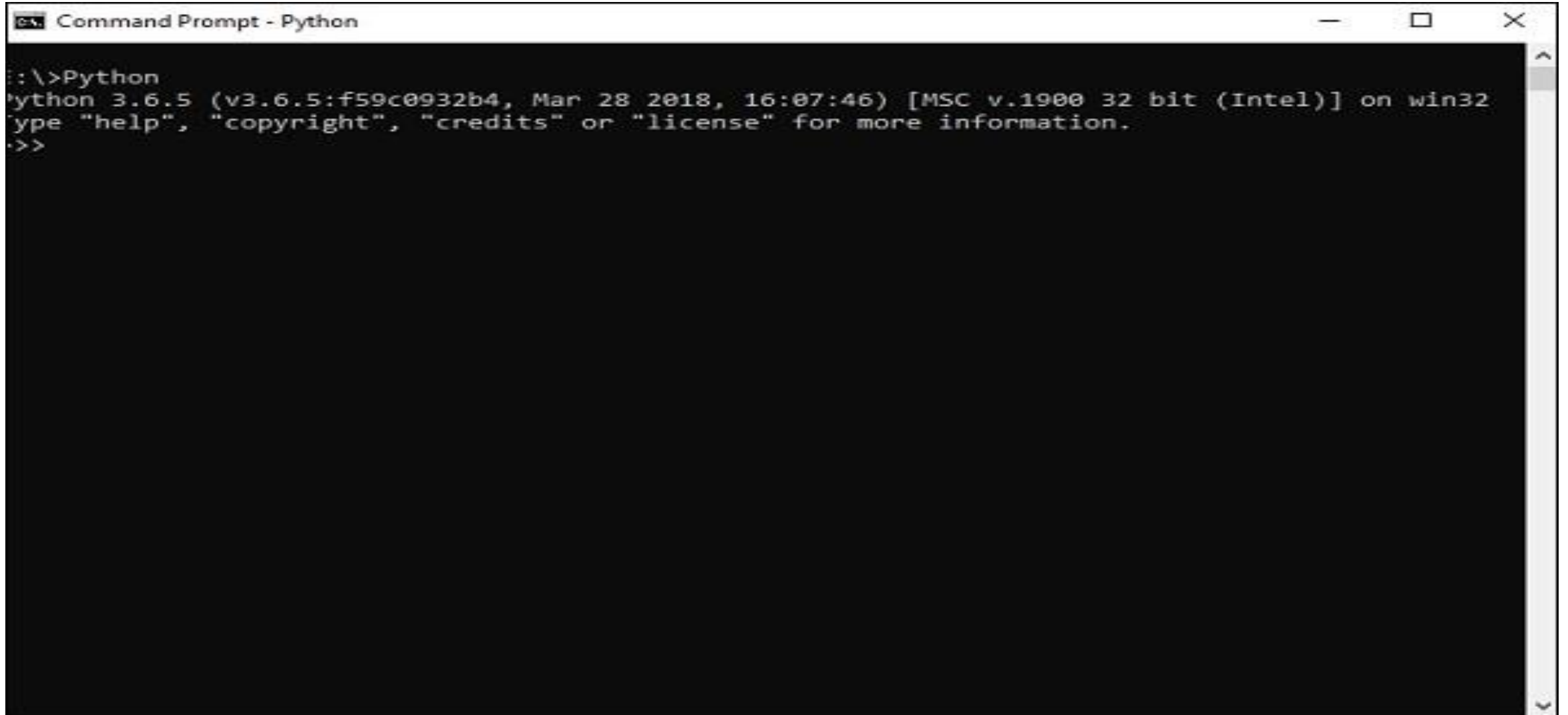


# *INSTALLING PYTHON PACKAGES*

# 1. INSTALL TENSORFLOW:

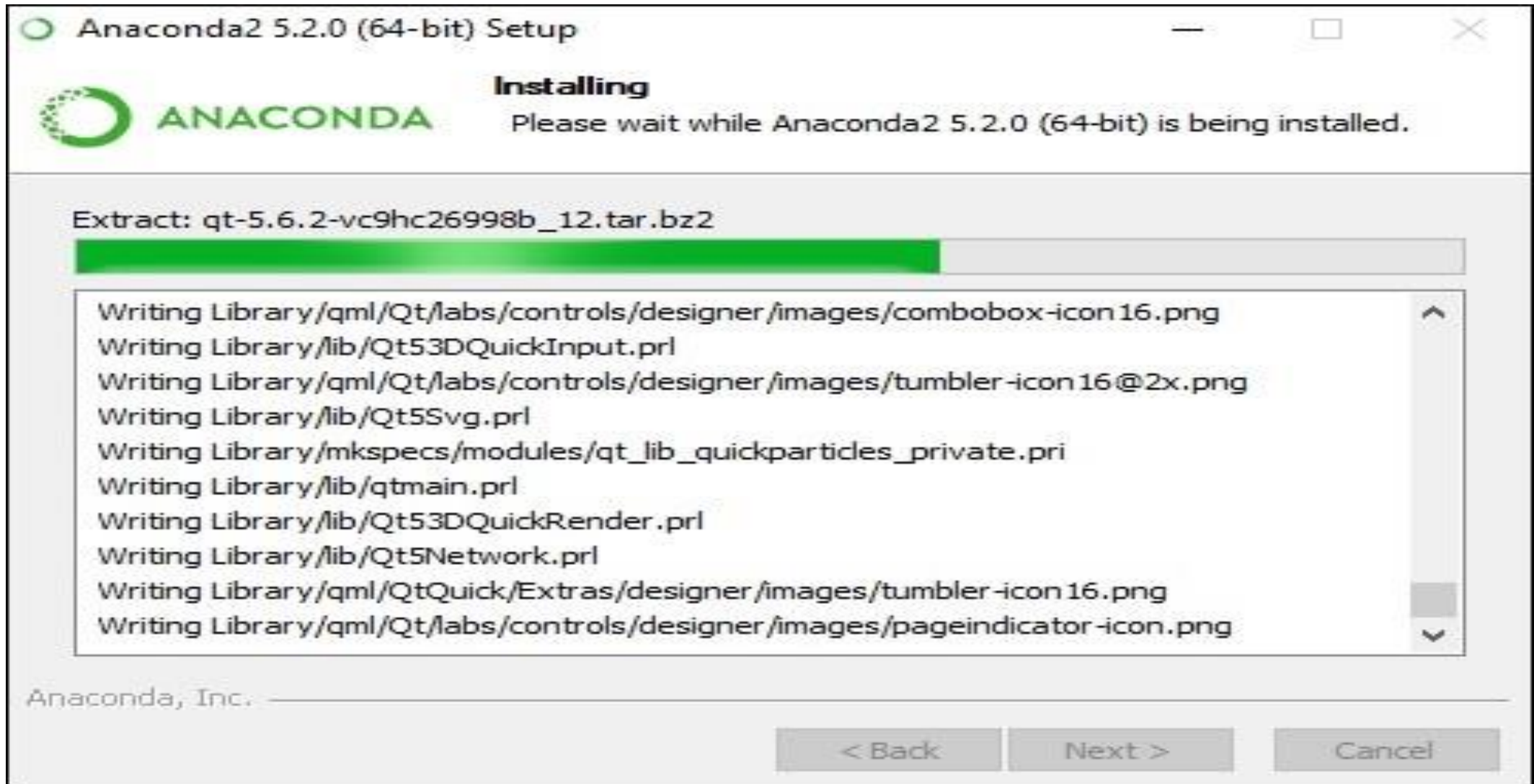
- Step 1 – Verify the python version being installed.



```
Command Prompt - Python

C:\>Python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>
```

## Step 2: Install Anaconda framework in our system.



# Step 3 – Execute the following command to initialize the installation of TensorFlow

**conda create --name tensorflow python=3.5**

```
Command Prompt - conda create --name tensorflow python=3.5

vc-14                h0510ff6_3                3 KB
wincertstore-0.2      py35hfebdb8_0            13 KB
wheel-0.31.1          py35_0                   81 KB
certifi-2018.4.16     py35_0                   143 KB
python-3.5.5          h0c2934d_2              18.2 MB
-----
Total:                20.8 MB

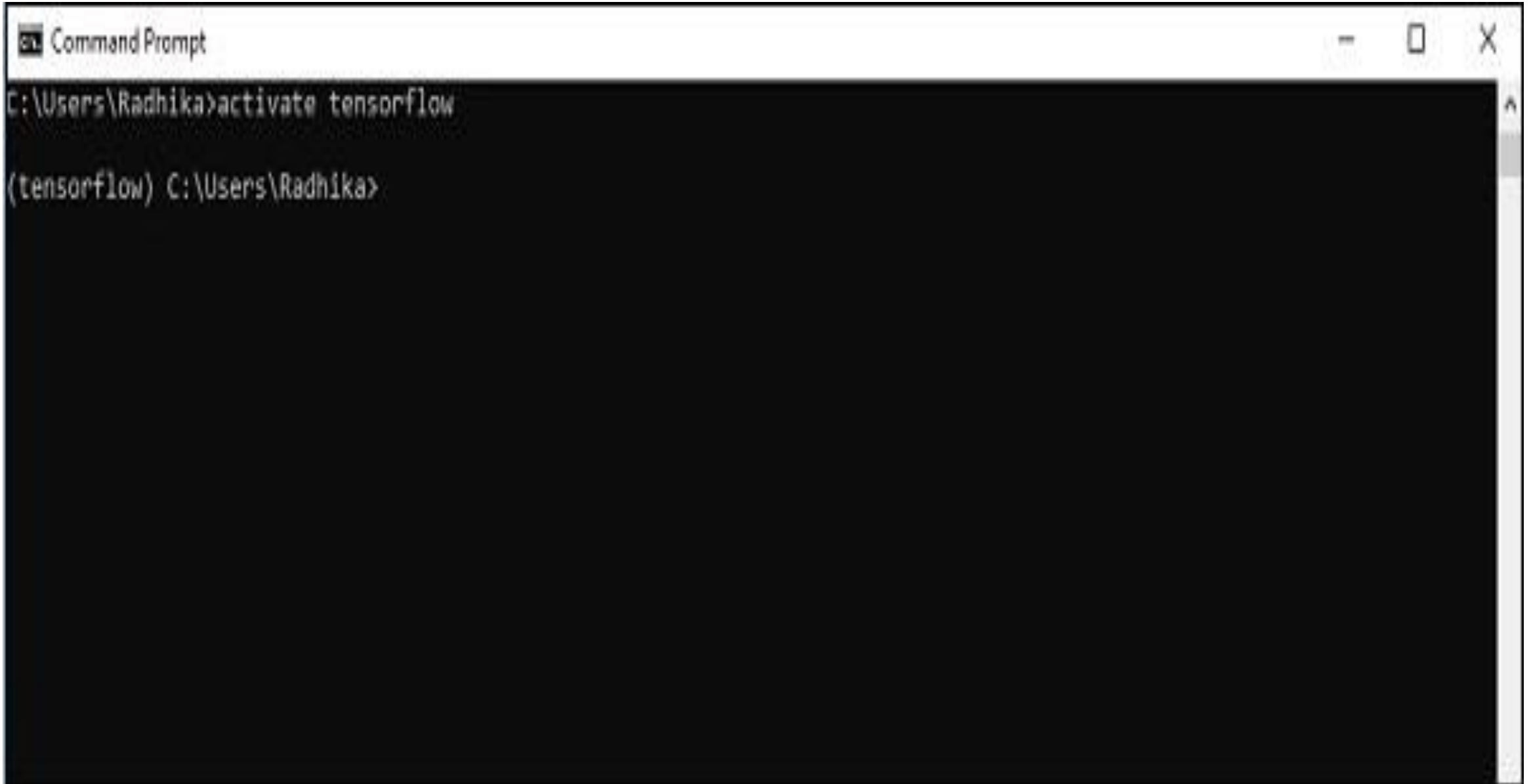
The following NEW packages will be INSTALLED:

certifi:      2018.4.16-py35_0
pip:          10.0.1-py35_0
python:       3.5.5-h0c2934d_2
setuptools:   39.2.0-py35_0
vc:           14-h0510ff6_3
vs2015_runtime: 14.0.25123-3
wheel:        0.31.1-py35_0
wincertstore: 0.2-py35hfdb8_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
pip-10.0.1                1.8 MB | ##### 100%
setuptools-39.2.0         593 KB | ##### 100%
vc-14                     3 KB   | ##### 100%
wincertstore-0.2          13 KB  | ##### 100%
wheel-0.31.1              81 KB  | ##### 100%
certifi-2018.4.16         143 KB | ##### 100%
python-3.5.5              18.2 MB| ##### 70%
```

## Step 4: ACTIVATE TENSORFLOW



```
Command Prompt
C:\Users\Radhika>activate tensorflow
(tensorflow) C:\Users\Radhika>
```



# Step 5 – Use pip to install “Tensorflow” in the system.

- **pip install tensorflow**
- **pip install tensorflow-gpu**

```
Command Prompt - pip install tensorflow
Requirement already satisfied: termcolor>=1.1.0 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: numpy>=1.13.3 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorflow) (1.14.5)
Requirement already satisfied: grpcio>=1.8.6 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: wheel>=0.26 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorflow) (0.31.1)
Requirement already satisfied: six>=1.10.0 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorflow) (1.11.0)
Requirement already satisfied: absl-py>=0.1.6 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorflow) (0.2.2)
Requirement already satisfied: astor>=0.6.0 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorflow) (0.6.2)
Requirement already satisfied: gast>=0.2.0 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: tensorboard<1.9.0,>=1.8.0 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorflow) (1.8.0)
Requirement already satisfied: setuptools in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from protobuf>=3.4.0->tensorflow) (39.2.0)
Requirement already satisfied: html5lib==0.9999999 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorboard<1.9.0,>=1.8.0->tensorflow) (0.9999999)
Requirement already satisfied: bleach==1.5.0 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorboard<1.9.0,>=1.8.0->tensorflow) (1.5.0)
Requirement already satisfied: markdown>=2.6.8 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorboard<1.9.0,>=1.8.0->tensorflow) (2.6.11)
Requirement already satisfied: werkzeug>=0.11.10 in c:\users\radhika\anaconda2\envs\tensorflow\lib\site-packages (from tensorboard<1.9.0,>=1.8.0->tensorflow) (0.14.1)
Installing collected packages: tensorflow
```

# “HELLO WORLD” IN TENSORFLOW :

```
C:\Users\Radhika>activate tensorflow

(tensorflow) C:\Users\Radhika>python
Python 3.5.5 |Anaconda, Inc.| (default, Apr 7 2018, 04:52:34) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, Tensorflow!')
>>> sess = tf.session()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: module 'tensorflow' has no attribute 'session'
>>> sess = tf.Session()
2018-06-28 11:12:04.586763: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
>>> print(sess.run(hello))
b'Hello, Tensorflow!'
>>>
```

## 2.Keras Installation Steps:

- **Step 1: Create virtual environment**
- **Virtualenv is used to manage Python packages for different projects.**

**For Windows using this command**

```
py -m venv keras
```



## Step 2: Activate the environment

FOR WINDOWS `\env\Scripts\activate`

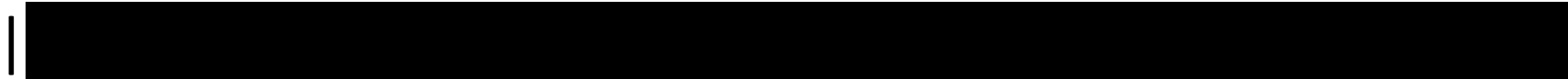
## Step 3: Python libraries

**NUMPY :**

`pip install numpy`

Collecting numpy Downloading

[https://files.pythonhosted.org/packages/cf/a4/d5387a74204542a60ad1baa84cd2d3353c330e59be8cf2d47c0b11d3cde8/numpy-3.1.1-cp36-cp36m-macosx\\_10\\_6\\_intel.macosx\\_10\\_9\\_intel.macosx\\_10\\_9\\_x86\\_64.macosx\\_10\\_10\\_intel.macosx\\_10\\_10\\_x86\\_64.whl](https://files.pythonhosted.org/packages/cf/a4/d5387a74204542a60ad1baa84cd2d3353c330e59be8cf2d47c0b11d3cde8/numpy-3.1.1-cp36-cp36m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.whl) (14.4MB)



## PANDAS: `pip install pandas`

- [illegible]

## MATPLOTLIB :pip install matplotlib

- Collecting matplotlib Downloading  
[https://files.pythonhosted.org/packages/cf/a4/d5387a74204542a60ad1baa84cd2d3353c330e59be8cf2d47c0b11d3cde8/matplotlib-3.1.1-cp36-cp36m-macosx\\_10\\_6\\_intel.macosx\\_10\\_9\\_intel.macosx\\_10\\_9\\_x86\\_64.macosx\\_10\\_10\\_intel.macosx\\_10\\_10\\_x86\\_64.whl](https://files.pythonhosted.org/packages/cf/a4/d5387a74204542a60ad1baa84cd2d3353c330e59be8cf2d47c0b11d3cde8/matplotlib-3.1.1-cp36-cp36m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.whl)  
(14.4MB)

| 14.4MB 2.8MB/s

# Keras Installation Using Python

- Pip install keras

**Quit virtual environment**

**DEACTIVATE**

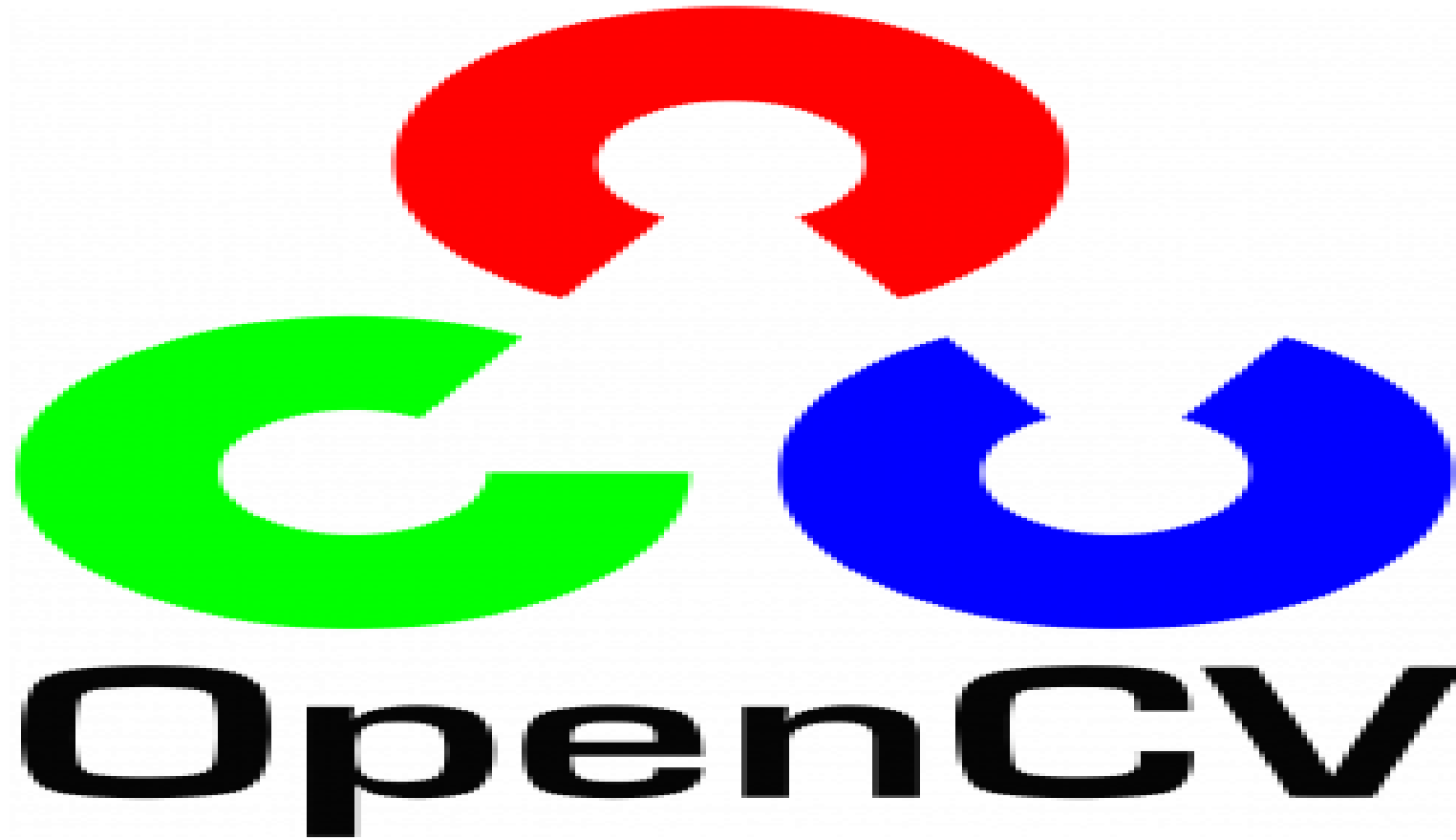
**INSTALL CONDA ENVIRONMENT**

**conda create --name PythonCPU**

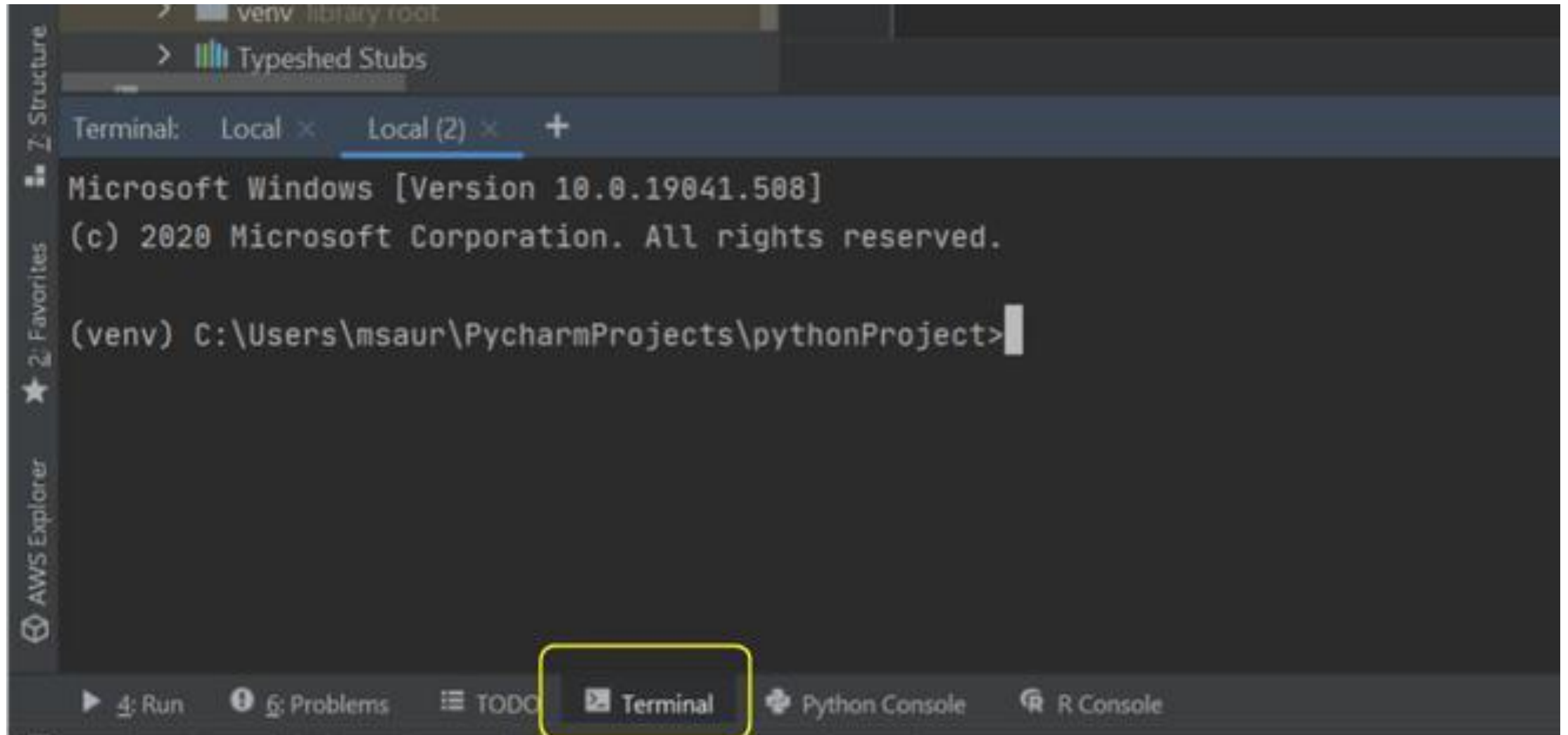
**INSTALL SPYDER :conda install spyder**

**conda install -c anaconda keras**

### 3.INSTALL OPEN CV

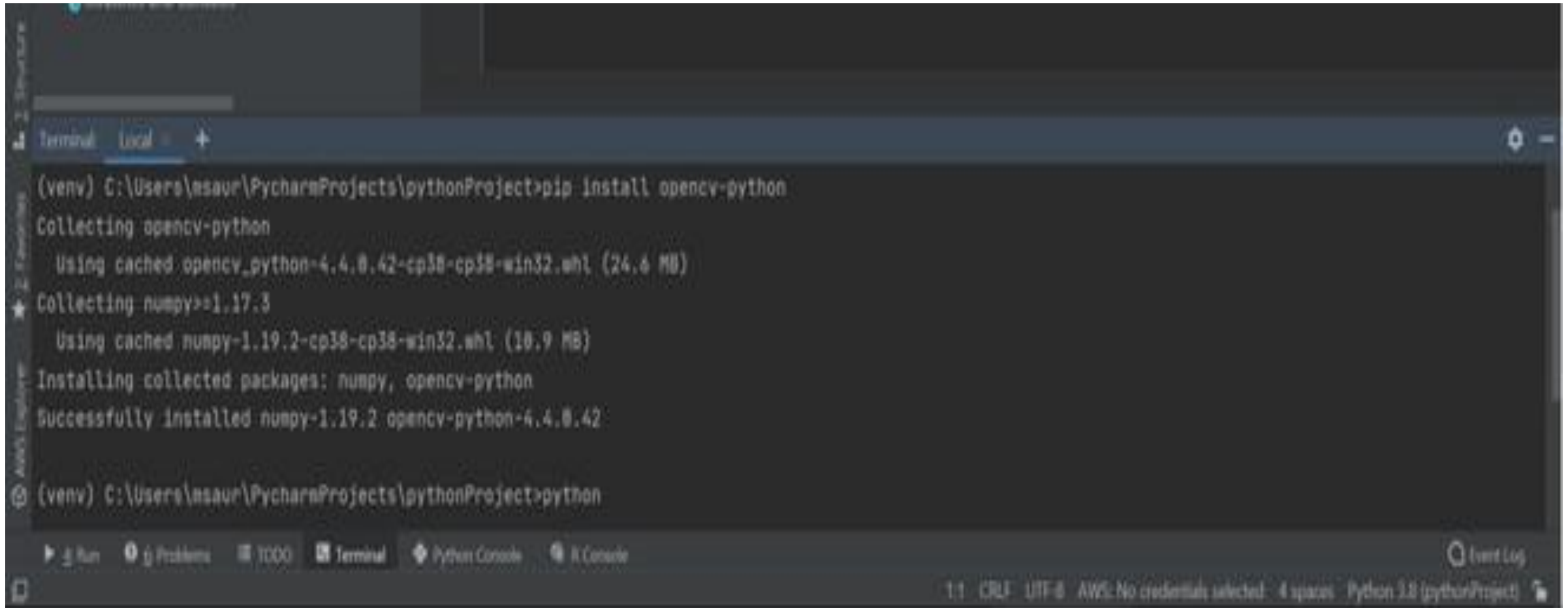


1) Go to the terminal option at the bottom of the IDE window





2) The pip (package manager) can also be used to download and install OpenCV. To install OpenCV, just type the following command:

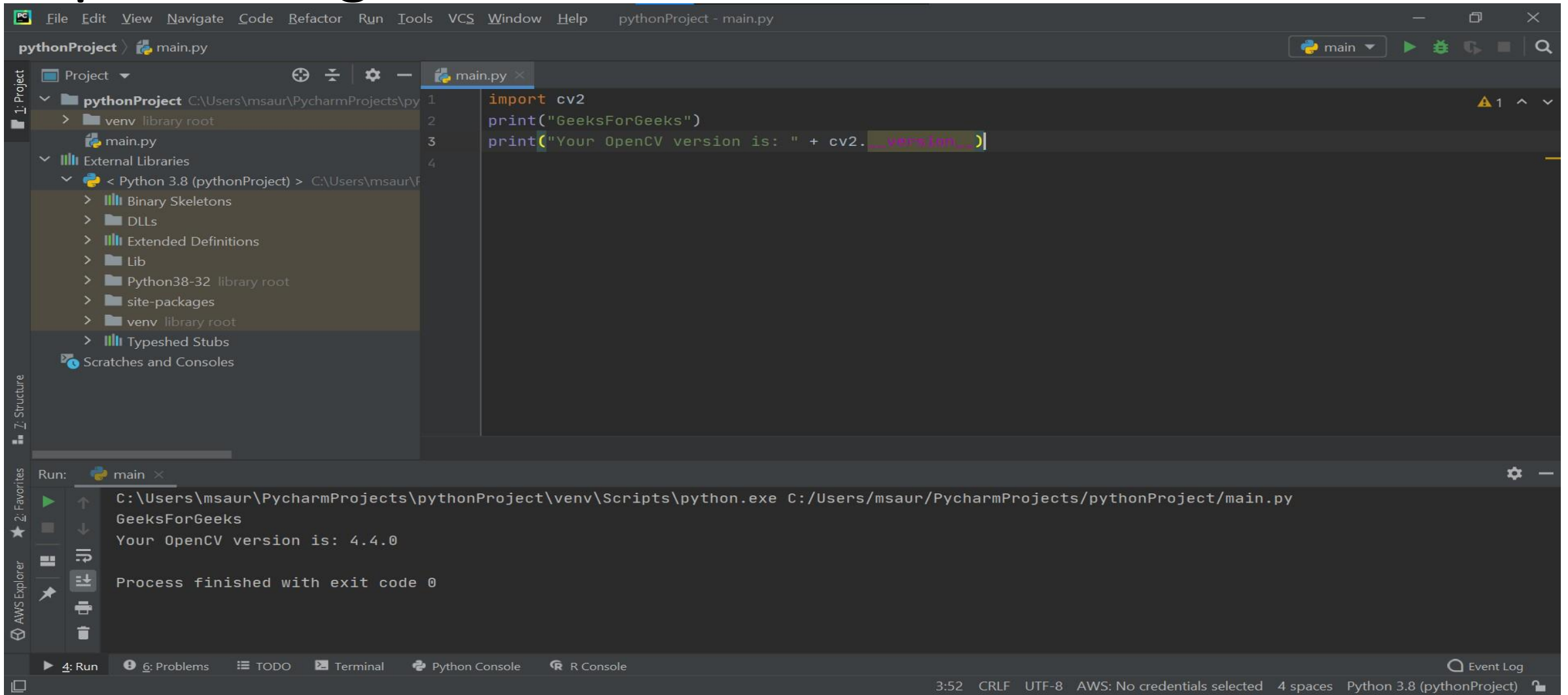


```
(venv) C:\Users\msaur\PycharmProjects\pythonProject>pip install opencv-python
Collecting opencv-python
  Using cached opencv_python-4.4.0.42-cp38-cp38-win32.whl (24.6 MB)
Collecting numpy>=1.17.3
  Using cached numpy-1.19.2-cp38-cp38-win32.whl (10.9 MB)
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.19.2 opencv-python-4.4.0.42

(venv) C:\Users\msaur\PycharmProjects\pythonProject>python
```

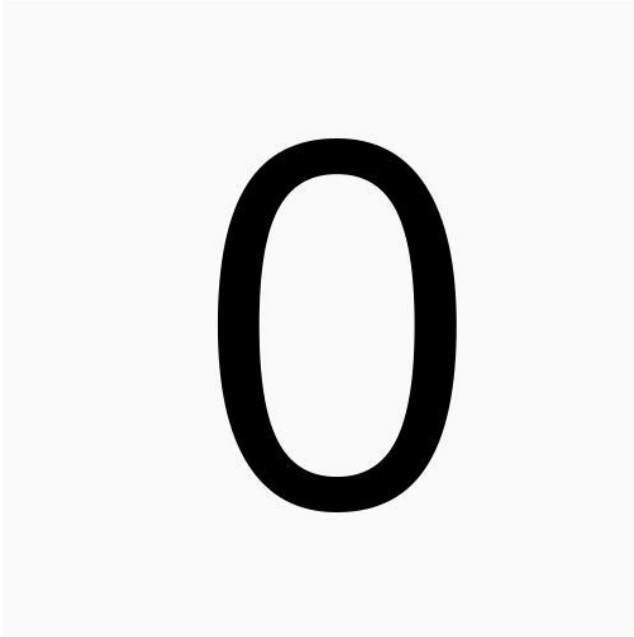
The screenshot shows a PyCharm IDE with a terminal window open. The terminal displays the command to install OpenCV and NumPy using pip. The output shows that the packages were successfully installed. The terminal window is titled 'Terminal' and 'Local'. The status bar at the bottom indicates the current file is '1:1 CRLF UTF-8 AWS: No credentials selected - 4 spaces Python 3.8 (pythonProject)'.

### 3) Now simply import OpenCV in your python program in which you want to use image processing functions.



### 3. pip install pillow

- Merging Two or More Images
- IMAGES USED

A large, bold, black digit '0' is centered within a light gray square. The square has a subtle gradient, being slightly darker at the edges. The digit '0' is a simple, clean, sans-serif style.

1

2

3

From PIL import Image

```
img_01 = Image.open("digit-number-img-0.jpg")
```

```
img_02 = Image.open("digit-number-img-1.jpg")
```

```
img_03 = Image.open("digit-number-img-2.jpg")
```

```
img_04 = Image.open("digit-number-img-3.jpg")
```

```
img_01_size = img_01.size
```

```
img_02_size = img_02.size
```

```
img_03_size = img_02.size
```

```
img_02_size = img_02.size
```

```
Print('img 1 size: ', img_01_size)
print('img 2 size: ', img_02_size)
print('img 3 size: ', img_03_size)
print('img 4 size: ', img_03_size)
```

```
new_im = Image.new('RGB', (2*img_01_size[0],2*img_01_size[1]),
(250,250,250))
```

```
new_im.paste(img_01, (0,0))
new_im.paste(img_02, (img_01_size[0],0))
new_im.paste(img_03, (0,img_01_size[1]))
new_im.paste(img_04, (img_01_size[0],img_01_size[1]))
```

```
new_im.save("merged_images.png", "PNG")
new_im.show()
```



OUTPUT

0

1

2

3

# CREATE A THUMBNAIL

- IMAGE USED



- From PIL import Image
- 
- # creating a object
- image = Image.open(r"image.jpg")
- MAX\_SIZE = (100, 100)
- 
- # Creating the thumbnail
- image.thumbnail(MAX\_SIZE)
- 
- image.show()

# OUTPUT

