

# Installing Python and OpenCV

## For Windows

### Step1: Download and Install Python

Go to the [python.org](https://python.org) website to grab our install. Go under the downloads menu and under the pop-up, you are going to click on the python version number to download. For this course, we are using python3.9 (However, it should be ok if you have a previous version of python 3 installed). This would then download as an .exe executable. You can open and run this .exe to start the process.

Here, you have a couple of things to keep in mind.

1. Check that the python version is 3.9
2. It comes with pip. This would be used to install other packages.
3. Make sure to tick the box that says: "Add Python 3.9 to PATH". If you don't do this, you won't be able to find python in the command line prompt.

Now click on install. If any warning messages come up, go ahead and click accept. Once we see that the setup was successful, we can close the window.

Next, we must check that python is installed correctly. Open up the command prompt and type:

```
python
```

This would open the python interpreter as shown below:

```
Python 3.9.6 (tags/v3.9.6:db3ff76, June 28 2021, 15:26:21)
[MSG v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Make sure that it is the version installed matches the 3.9.subversion that you used to download from the python.org website, else you have a clash on your system.

Now we need to add some of the missing packages that we need for this course.

You can then type `exit()` to exit the python console.

```
>>> exit()
```

## **Step2: Create Virtual Environment**

```
mkdir opencv_for_beginners
```

```
cd opencv_for_beginners
```

This has created an empty folder. Let us now create a virtual environment inside this folder. We would use the command:

```
python -m venv opencv-env
```

This would generate another folder representing our environment.

```
dir opencv-env
```

Using `dir`, we can see a few subfolders inside this. We are most interested in the `Scripts` folder.

```
dir opencv-env\Scripts
```

This contains an `activate` file, which must be executed to activate our environment.

If you don't do this your installs would be system-wide. We highly recommend isolating your installs using a virtual environment.

We'll activate the environment

```
.\opencv-env\Scripts\activate
```

This command will activate our virtual environment as we can see our virtual environment name as a prefix on the command line.

```
(opencv-env)
```

This tells us that everything that we run and install, takes effect in our virtual environment.

### **Step3: Install Packages**

We will use pip package manager to install all packages. You can install any package by simply typing: "pip install" followed by the name of the package. You can install multiple packages simultaneously by typing on the same line. If you run into any errors, we would recommend following up by installing one at a time and also upgrading pip itself.

The key package to install is:

```
pip install opencv-contrib-python streamlit jupyter moviepy ipykernel  
matplotlib
```

This would install all the mentioned packages.

Now lets open python interpreter and import OpenCV

```
python  
  
>>> import cv2  
  
>>> cv2.__version__
```

We can see that this does not give an error and we have successfully installed OpenCV.

Now exit the interpreter.

```
>>> exit()
```

## For Ubuntu

### Step1: Update Package information

```
$ sudo apt-get update
```

You will then need to enter your password

### Step2: Install Python 3.9

```
$ sudo apt install python3.9
```

Press yes to continue.

After this is done we would check the version of python:

```
$ python3.9 --version
```

Python 3.9.5

### Step3: Setup Python Virtual Environment

Let's setup our virtual environment by first

```
$ sudo apt install python3.9-venv
```

Press yes to continue.

### Step4: Create Virtual Environment

```
$ python3.9 -m venv ~/opencv-env
```

This creates a virtual environment named opencv-env in the user's home directory.

You can also note that you can use any name for your virtual environment but we re also recommending that you use opencv-env for consistency within this course.

### Step5: Activate/ Deactivate the Virtual Environment

Let's see how to activate our virtual environment.

```
$ source ~/opencv-env/bin/activate
```

This command will activate our virtual environment as we can see our virtual environment name as a prefix on the command line.

```
(opencv-env)~$
```

### **Step6.1: Install pip**

We will use pip to install various packages in our virtual environment. First, we must confirm that pip is installed.

```
(opencv-env)~$ which pip
```

This command will tell us which pip version is installed in our system. We should also check whether we have any updates available for pip. For that we would use the following command:

```
(opencv-env)~$ pip install --upgrade pip
```

So at this point, we are ready to use pip to install OpenCV and several other packages.

### **Step6.2: Install OpenCV using pip**

For this, we would use the following:

```
(opencv-env)~$ pip install opencv-contrib-python
```

Note that we are using the contributor version of OpenCV as it has several more classes and functionality than the base OpenCV.

### **Step6.3: Install Matplotlib**

For installing matplotlib, we would use,

```
(opencv-env)~$ pip install matplotlib
```

### **Step6.4: Install Streamlit**

Next, we would continue on for streamlit:

```
(opencv-env)~$ pip install streamlit
```

Now you may find that streamlit would not install correctly at first. If you run into this, we are going to run another command to install some additional dependencies on your machine.

```
$ sudo apt install python3.9-dev
```

Press yes to continue.

And then we will re-run our install streamlit command. If you are still running into issues, it means you do not have gcc installed on your system.

```
$ sudo apt install gcc
```

Press yes to continue.

And then we will re-run our install streamlit one more time.

### **Step6.5: Install Moviepy**

```
(opencv-env)~$ pip install moviepy
```

### **Step6.6: Install Ipykernel**

For use with our jupyter notebooks we would install ipykernel:

```
(opencv-env)~$ pip install ipykernel
```

### **Step6.7: Install PyAutoGUI**

Next, we would install PyAutoGUI:

```
(opencv-env)~$ pip install pyautogui
```

### **Step6.8: Install Mime**

Followed by mime:

```
(opencv-env)~$ pip install mime
```

### Step6.9: Install Mediapipe

```
(opencv-env)~$ pip install mediapipe
```

### Step7: Verify our installs

Now let's check that OpenCV was correctly installed in our system. We can do this by entering a python console by typing:

```
(opencv-env)~$ python3.9
```

This opens up the python console which looks like this:

```
>>>
```

Now we will import OpenCV:

```
>>>import cv2
```

Once we run this, we can also check the version of OpenCV by running:

```
>>>cv2.__version__
```

You can then type `exit()` to exit the interpreter.

```
>>> exit()  
(opencv-env)~$
```