

General Python Installation Instructions:

These are general instructions and guidelines that are applicable for all workshops. Kindly read these before you move on to instructions for specific workshops.

Important:

1. We recommend that you follow the instructions given for the workshops that interest you and install the necessary software and packages **before you come** to the conference.
2. We **strongly recommend** installing Anaconda with Python 3.7 since it is convenient, hassle-free way to install Python, all the required packages, Package manager (PIP), a simple Text Editor and a development environment all in one go.

Install Python:

We recommend using Python 3.6 or above. You can find the links to the installers given below;

For Windows & Mac: <https://www.python.org/downloads/release/python-368/>

For Linux: Follow the instructions provided here, <https://docs.python-guide.org/starting/install3/linux/>

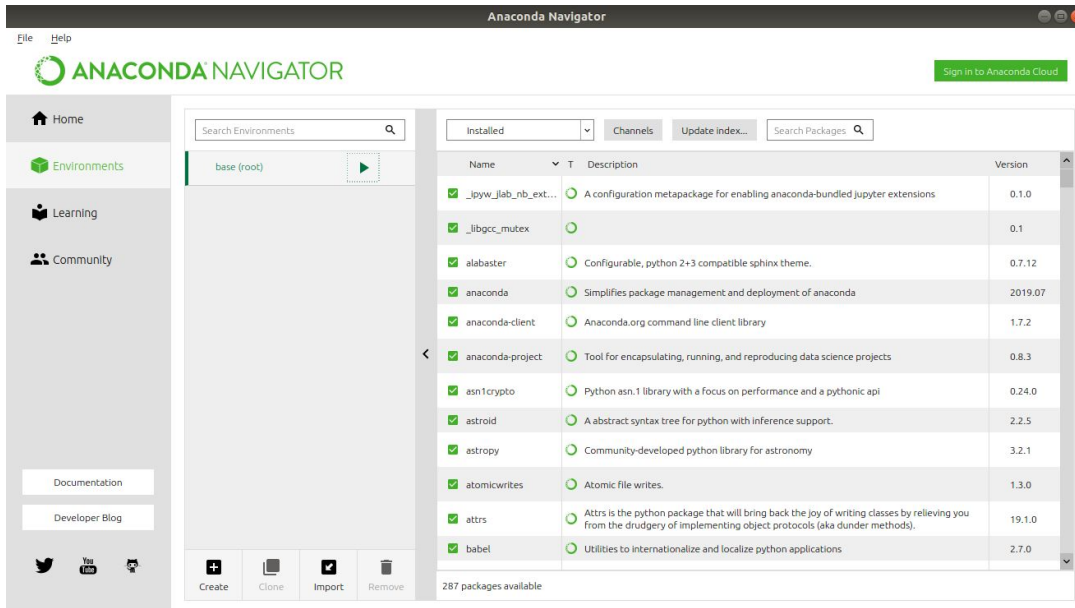
Certain workshops require Python 3.7. In case your system already has Python 3.7 installed, you are good to go. If not, we recommend that you install Anaconda for your OS. Please see **Installing Anaconda** instructions

Installing Anaconda:

1. Most of the workshops require you to install Python & Python Scientific Computation packages (Numpy, Matplotlib, Pandas, Scipy, Numba etc.).
2. We **strongly recommend** installing Anaconda with Python 3.7 since it is convenient, hassle-free way to install Python, all the required packages, Package manager (PIP), a simple Text Editor and a development environment all in one go.
3. Go to <https://www.anaconda.com/distribution/> and install the Python 3.7 version for your Operating System.
4. Once installed, Anaconda comes with all these packages [pre-installed](#).

How to install additional packages in Anaconda:

In case you prefer to install, upgrade a Python package manually when using Anaconda, you can follow the instructions below:



1. Open the Anaconda Command Prompt with administrator privileges. Open **Anaconda Launcher/Navigator** >> Click on **Environments** Tab >> Click on the Arrow beside Base >> Click on **Open Terminal**
2. Use the commands below:
 - a. In order to install a package: *pip install <package_name>*
 - b. In order to install a package of a particular version: *pip install <package_name>==<version_number>* For Example: *pip install numpy==1.11.4*
 - c. In order to remove a package: *pip uninstall <package_name>*

Install PIP:

PIP is a Python Package manager that allows you to easily install third party Python packages

Install on Windows: <https://www.liquidweb.com/kb/install-pip-windows/>

Install on Mac & Linux: <https://www.makeuseof.com/tag/install-pip-for-python/>

Setting up Ubuntu in a Virtual Machine:

You can follow the instructions provided here: <https://itsfoss.com/install-linux-in-virtualbox/> to setup a virtual machine with the Ubuntu OS.

Things to Remember:

1. On Windows I get the error **“ImportError: DLL load failed”**, what to do?
Solution: On Windows systems in case you encounter the error **“ImportError: DLL load failed”** during the usage of Canopy, please follow the instructions and

install the Visual C++ Redistributable for Visual Studio 2015 from the link given below;

<https://www.microsoft.com/en-US/download/details.aspx?id=48145>

2. Which installer should I use *32 bit* or *64 bit* ?

Solution: The type of installer (32 or 64 bit) depends on the type of Operating System and Hardware you use. If your Operating System is 64 bit, it will require the 64 bit installer, if it's a 32 bit System, you should download the 32 bit installer.

You can check whether your system is a 32 bit or 64 bit one by following the steps listed in the posts, below:

For Windows:

<https://www.howtogeek.com/howto/21726/how-do-i-know-if-im-running-32-bit-or-64-bit-windows-answers/>

3. On Linux when installing Matplotlib manually, please ensure that you have installed the Tkinter system package;
- a. *On Debian based system:* **sudo apt-get install python3-tk**
 - b. *On RPM based system:* **sudo yum install tkinter**

KuttyPy: Learn Microcontrollers with Python (Basic)

Speaker: Jithin B.P.

A detailed tutorial and installation instructions for KuttyPy can be found on the following [page](#)

Requirements:

- Participants must bring their own laptops, with Ubuntu 18.04+ installed. Windows 7/10 is also supported. Other Linux OSes are also supported via Pip, but please go through the installation process beforehand in order to be sure.
(<https://kuttypy.readthedocs.io/en/latest/workshop/>)
- python3 must be pre-installed, along with pip
- For ubuntu, the deb file will fetch all dependencies (python3, pyqt5, pyserial, scipy, numpy, avr- gcc, qtconsole). Please refer to <https://kuttypy.readthedocs.io/en/latest/workshop/> which has detailed instructions and the download link.
- For Windows, pip3 can be used to install the KuttyPy package. Winavr, USB driver must be downloaded and installed. Instructions <https://kuttypy.readthedocs.io/en/latest/workshop/>
 - A bundled installer(70MB approx) is also available which installs everything, but is not the preferred route because it uses pyinstaller, and Python3 is not installed globally.
 - Participants may optionally bring electronic components they wish to try and control using Python scripts such as LEDs, LDRs, Relays, Small motors, Resistive sensors etc.

Robotics programming with rospy (Basic)

Speaker: Mohit Khandelwal, Akshita Kanojia

Prerequisites:

- Python 3.x installed in local machine
 - ROS Kinetic installed in local machine
- OR**
- Account on web browser based ROS Development Studio

Learners are expected to have prior python programming experience. A basic understanding of robotics will also be helpful.

Robot Operating System (ROS) setup instructions for Ubuntu 16.04 linux environment:

The following set of commands will add ROS Kinetic to system list of software sources, download and install the ROS packages and set up environment and ROS build tools -

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyervers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
```

```
sudo apt-get update
```

```
sudo apt-get install ros-kinetic-desktop-full
```

```
sudo rosdep init
```

```
rosdep update
```

```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
```

```
sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-essential
```

To verify the installation is successful use the following command

```
rosversion -d
```

If you are getting 'kinetic' as the output, you are all set with installation.

You can visit <http://wiki.ros.org/kinetic/Installation> for setup instructions on other operating system.

Installation For Ubuntu 18.04: Follow the instructions provided here.

<http://wiki.ros.org/melodic/Installation/Ubuntu>

Installation For Other Platforms: Please follow the instructions given here after clicking on the correct Operating System that you use, <http://wiki.ros.org/melodic/Installation>

Alternatively you can do ROS development on a single web interface using cloud based ROS Development Studio. To access it in your web browser visit

<https://www.theconstructsim.com/rds-ros-development-studio/> . Create a free account and you are ready to go.

Simulation of wireless communication system using Python (Intermediate/Advanced)

Speaker: Ashok Govindarajan

Requirements:

- Python 3.5.3 or above to be installed
- Packages:
 - Matplotlib 2.2.3 (pip3 install matplotlib==2.2.3)
 - Scipy 1.1.0 (pip3 install scipy==1.1.0)
- Install Tkinter - On Linux using the command **sudo apt-get install python3-tk**

Note: Instead of installing the above packages individually, Windows users can instead install the Anaconda IDE

1. Download the Anaconda installer from here: <https://www.anaconda.com/distribution/>
2. Follow the instructions provided [here](#)

After installing the above-mentioned packages, download [this](#) python script and run it using the command “**python 16_QAM.py**” to check if your installation is successful. The participants can run the given code and check if they are able to get 5 plots and a BER around 0.004545

Calculus with Python (Basic)

Speaker: Nishadh K.A

Requirements:

- Python 3
 - Numpy (pip3 install numpy)
 - Scipy (pip3 install scipy)
 - Sympy (pip3 install sympy)
 - Matplotlib (pip3 install matplotlib)
 - Pyclaw (pip3 install clawpack)
 - Jupyter Notebook (pip3 install jupyterlab)

Installation Note:

For Linux:

Before installing Pyclaw, please ensure that you have GFortran (Fortran compiler) installed. If not, you can run the command **sudo apt-get install gfortan** (for debian systems) and **sudo yum install gcc-gfortran** (for RPM based system like Fedora/CentOS)

For Windows:

In case you face issues, You can install MinGW and add the binaries to the PATH variable

Note: In case the participants are not able to set up the above libraries on the day of the workshop, a Jupyter Hub instance of workshop material will be hosted. Participants are requested to arrange their own internet connection to access it. The workshop material will be available at <https://github.com/nishadhka/Learning-Calculus-with-Python>. It is suggested to go through the workshop material before the actual workshop, in case if there are any new updates.

Animating Science using Python (Basic)

Speaker: Purusharth Saxena, Sharanya Achut

Installation instructions for the workshop

For help with the installation, join the slack [channel](#).

Windows 10

Please follow the [video](#) to install manim on Windows machine.

The installation is divided into three parts

1. Installing System Libraries
2. Installing manim's python Dependencies
3. Testing the Installation

1. Installing system libraries

The required system libraries are:

1. Python 3.7
2. ffmpeg
3. Latex
4. Sox

Installing Python

Download Python: <https://www.python.org/ftp/python/3.7.4/python-3.7.4-amd64.exe> and install it (don't change any of the default options).

Installing ffmpeg

Ffmpeg is used for video, audio, and other multimedia files and streams and is the de-facto open source library for the video rendering/processing. Download ffmpeg:

<https://ffmpeg.zeranoe.com/builds/win64/static/ffmpeg-20190826-0821bc4-win64-static.zip>

Extract the files in C:\ffmpeg (You will have to create the folder ffmpeg inside C directory)

Note: Typically, the folder for ffmpeg will contain a subfolder named ffmpeg-2019826-.... , cut the contents from inside this folder so that C:\ffmpeg\bin\ is a valid path.

Installing MikTex

Download MikTeX from

<https://miktex.org/download/ctan/systems/win32/miktex/setup/windows-x64/basic-miktex-2.9.7152-x64.exe> and install it with the default options.

Install SoX

Download SoX from: <https://sourceforge.net/projects/sox/files/sox/> and install it with the default options.

Adding to PATH

Adding to PATH is important so that python, and in turn, manim knows where to look for all the files you just installed. Go to This PC > Right Click > Properties > Advanced System Settings > Environment Variables . Under "**User Variables for Admin**", click on Path and press the edit button. Copy Paste the following after pressing new on the pop-up window (you'll have to paste one at a time):

```
%localappdata%\Programs\Python\Python37\  
%localappdata%\Programs\Python\Python37\Scripts\  
%localappdata%\Programs\MiKTeX  
2.9\miktex\bin\x64\ C:\ffmpeg\bin\
```

After you're done, open up terminal and type python. If you get the python command prompt, that means the installation has been successful (so far). Similarly, type ffmpeg , if you see the ffmpeg version number, then you're good to go. (In any case if you get an error saying '...' is not recognized as an internal or external command , then something went wrong somewhere and it'll be a good idea to retrace your steps before proceeding further).

2. Installing manim Dependencies

Before we install other manim dependencies, we need to download and install Microsoft C++ Redistributable Library from:

<https://www.microsoft.com/en-US/download/details.aspx?id=48145>. These are needed for the scipy library.

Once that is done, download PyCario from: <https://www.lfd.uci.edu/~gohlke/pythonlibs/#pycairo>
Refer the image below to make sure you've downloaded the correct version.

Finally, navigate to Downloads from the command prompt and do:

```
pip install pycairo-1.18.1-cp37-cp37m-win-amd64.whl
```

Or

```
pip install pycairo-1.18.1-cp37-cp37m-win32.whl
```

After that, in the command prompt type:

```
pip install --upgrade setuptools
pip install pyreadline
pip install manimlib
```

3. Testing the installation

Once that is done, in the same command prompt, type: `manim -h` If you see something similar to what is shown in the image, it means you're ready for the workshop

Mac OS X Mojave

While installing dependencies on Mac, it'll be more convenient to install the dependencies through homebrew . The installation procedure remains the same except, you'd have to install homebrew if you don't have it already.

Installing Homebrew

Homebrew is a package manager for Mac, it's equivalent to `apt/dnf/yum` in Ubuntu and Fedora. Open up terminal and paste the following command:

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

1. Installing System Libraries

Cairo, SoX, ffmpeg

Once the installation for homebrew is completed, use the same terminal to type out the following command one after another:

```
brew install cairo
brew install sox
brew install ffmpeg
```

Installing Latex

Download and install MiKTeX from

https://miktex.org/download/ctan/systems/win32/miktex/setup/darwin-x86_64/miktex-2.9.7050-1-darwin-x86_64.dmg

This would be a .dmg file. Drag and drop it to the Applications folder to complete the installation. (For more information on installation instructions: <https://miktex.org/howto/install-miktex-mac>))

Installing Python3

Note that all Macs come with Python2 pre-installed. You'll have to explicitly install Python3
Download and install python 3.7.4 from the python.org website:

<https://www.python.org/ftp/python/3.7.4/python-3.7.4-macosx10.9.pkg>

2. Installing manim

In the terminal type:

```
pip3 install manimlib pycairo
```

Testing the installation

Type `manim -h` on the terminal and check if you get a help menu.

Common Errors:

If you get an error saying `ModuleNotFoundError: No module named 'cairo'` Check if you have `pkg-config` installed.

Open a new terminal windows and do the following:

1. Install package config: `brew instal pkg-config`
2. Export path:

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig"  
export LDFLAGS="-L/usr/local/opt/libffi/lib"
```

Test the installation again and check. If you see a manim help screen, then you're ready for the workshop.

Linux (Ubuntu/Fedora)

Ubuntu

```
sudo apt install libcairo2-dev ffmpeg sox texlive-full
```

```
python3 -m pip install manimlib
```

Fedora

```
sudo dnf install cairo-devel ffmpeg sox texlive-scheme-full
```

```
python3 -m pip install manimlib
```

Note: If you want to reduce the size of the installation, [download and install MikTeX](#). You can drop the texlive-full or texlive-scheme- full if you decide to go for the MiKTeX installation.

Testing the installation

Type `manim -h` on the Terminal. If you see the help message, then you're ready for the workshop.

Application of Machine Learning in Medical Image Analysis (Intermediate/Advanced)

Speaker: Tanmoy Bandyopadhyay

Prerequisites:

Preferably participants should be having basic knowledge of Machine Learning (like regression vs. classification, labels, features, loss function, accuracy, training/test/validation set).

Following link can be referred for an overview of Machine Learning:

<https://developers.google.com/machine-learning/crash-course/ml-intro>

Following link can be referred to get an overview of “scikit-learn” python module for machine learning:

<https://scikit-learn.org/stable/tutorial/basic/tutorial.html>

(It will sufficient to understand the basic steps of using machine learning for classification task like loading data, splitting data in training/test set, getting a model, train (fit) a model, predict using the trained model, measure accuracy.)

A basic overview of “Bag-Of-Words” is also recommended.

https://en.wikipedia.org/wiki/Bag-of-words_model_in_computer_vision

Software to be downloaded and installed

Anaconda3-5.2.0-*** (the suffix *** depends on your OS) from the link given below:

<https://repo.continuum.io/archive/>

After getting Anaconda distribution of Python install “tensorflow” followed by “keras”. Refer the following links:

<https://anaconda.org/conda-forge/tensorflow>

<https://anaconda.org/conda-forge/keras>

Interfacing Python with Javascript (Intermediate/Advanced)

Speaker: Michael Droettboom

Prerequisites:

- A modern web browser
- The latest release of Firefox (preferred for performance) or the latest release of Chrome. Safari, Edge and Internet Explorer will not work.

This workshop will explore how Pyodide can be used to build Python applications that interact with Javascript and Web APIs.

Two worked examples will be explored as a group:

- Building a simple browser-based painting application
- Optimizing the k-means algorithm in Python by replacing key parts with Javascript

Applied Machine Learning in Python using scikit-learn, mlxtend and pandas
(Intermediate/Advanced)

Speaker: Ashita Prasad

Software requirements:

- Python 3
 - pandas (pip3 install pandas)
 - mlxtend (pip3 install mlxtend)
 - Scikit-learn (pip3 install scikit-learn)
 - Matplotlib (pip3 install matplotlib)

Workshop: Accelerating Data Science & Scientific Computing in Python using Numba (Intermediate/Advanced)

Speaker: Ankit Mahato

Requirements:

- **Python 3.4+**
 - Matplotlib (pip3 install matplotlib)
 - Pillow (pip3 install pillow)
 - Numba (pip3 install numba)
 - Scikit-learn (pip3 install scikit-learn)
 - Pandas (pip3 install pandas)

Note: On Linux systems, make sure that you have Tkinter installed

Digital Signal Processing using Python (Intermediate/Advanced)

Speaker: R Senthil Kumar

- Python3.7
 - Scipy (pip3 install scipy)
 - Numpy (pip3 install numpy)
 - Matplotlib (pip3 install matplotlib)

Note: On Linux systems, Make sure that you have Tkinter installed

Building ML Applications with Gramex (Intermediate/Advanced)

Speaker: Kamlesh Jaiswal, Nivedita Deshmukh

The prerequisites for the Gramex ML workshop are here:

<https://github.com/gramexrecipes/gramex-ml-workshop/blob/master/README.md>

Building an open source voice assistant with Python deepspeech library by Mozilla (Intermediate/Advanced)

Speaker: Ranjith Raj Vasam

- Python3
- pip3
- npm
- GNU/Linux system(preferable, but not mandatory)