Getting started with Python

Installation

conda

To get started with Python it is recommended to install an environment manager. An environment manager lets you create a virtual box within you system in which you can install a variety of python dependencies and packages without disturbing, or interacting with the main system. These environments can be used with Python, R, Java etc. One of the most common ones is conda. This can be installed following these instructions:

conda installation

There is a small difference between Miniconda and Anaconda, in which Miniconda is a minimal version of Anaconda. It is advised to use Miniconda given that it doesn't contain any unnecessary packages.

After installing conda on your system we can create a new environment. You can do so by running the following command in your terminal:

```
conda create -n nameofyournewenvironment python=3.7
```

Once the environment has been created it can be started by executing this command, likewise in a terminal:

```
conda activate nameofyournewenvironment
```

pip

Pip is a package installation helper that allows you to easily install packages within your conda environment. Alternatively you can also use conda directly once you have activated you environment in a terminal.

To install pip run the following command once after activating the environment (see Section: conda):

```
conda install pip
```

Now you can use pip as well as conda to install further packages that might be useful or required.

This is generally done by executing:

```
conda install nameofpackage
```

OpenCV

OpenCV is an open-source, computer vision package that has many useful functions to offer, especially when working with images and videos.

You can install OpenCV once you have installed pip (see Section: pip) using the following command within your activated environment:

```
pip install opencv-python
```

This allows you to use all functions that are implemented in OpenCV and well documented!

Jupyter

Jupyter allows you to use browser based notebook structure. This means that you can create notebooks composed of cells that can be code based or text (Markdown). The code can be written and interpreted in various different coding languages, although these must be defined by specifically installed ipython kernels. Here we will focus on using jupyter notebooks with python.

To install jupyter run the following command once you have activated your environment in a terminal (see Section: conda):

```
pip install jupyter
```

You can now start using jupyter notebooks! In order to get started with a new notebook you have to initiate jupyter with the following command from within your environment, once jupyter is installed:

```
jupyter notebook
```

This should automatically open a window in your browser, and visualize the jupyter interface. Here you can create new notebooks by clicking on the ‘New’ tag and selecting the ‘Python 3’ Kernel.

In order to use an already created notebook you must change directory prior to executing the `jupyter notebook` command. The directory to change to should contain the notebook you want to execute or be above it. If you `cd` into `Desktop/working_folder` you cannot access files that are within `Desktop`. However, if you `cd` into `Desktop` you can access files on the `Desktop` and within the `working_folder`.

Within a jupyter notebook you can choose the cell type by selecting ‘Cell’ > ‘Cell Type’. ‘Code’ refers to code and ‘Markdown’ lets you write normal, human-readable text.