Computational Motor Control - 2019

February 21, 2019

Contents

Student names: . . . (please update)

Instructions: This document contains the instructions to install and get familiarized with Python programming. ***This lab is not graded.*** *This file does not need to be submitted and is provided for your own benefit.*

1 Course Organization

This year all the exercises for the course will be distributed through EPFL's GitLab. GitLab is an online open source Git repository for maintaining and distributing code between groups of people. The access to the exercises repository is public and you can directly use your epfl credentials to login. We recommend you to use the features Git offers to collaborate with your partners and maintain a history of your own work as well. If you download the repository without cloning, then you will have to do it every week and you will end up downloading all the previous files every time you do it. This report does not contain a tutorial on how to use Git but you can find the references in section 7.2. However, we will have short demo on how to use it during the exercise hour.

The link to the git repository

This course expects you to complete the exercises using the standard Python programming language. There are many options to use Python. For those students who are starting of new with Python, we strongly suggest to follow the following installation steps. **NOTE :** *All the Python codes provided in this lab should be studied carefully as these basic notions will be needed in the future exercises and projects of this course.*

NOTE : *The course supports the use of Python 3 unless otherwise mentioned.*

2 Windows

2.1 Python

2.1.1 Checking if python exists

Before installing please check if you already have python installed on your computer. To do so open Git Bash (Application you must have already installed while following the Git Instructions document)

Once Git Bash is open execute the following commands,

```
$ python -V
```

```
$ python3 -V
```

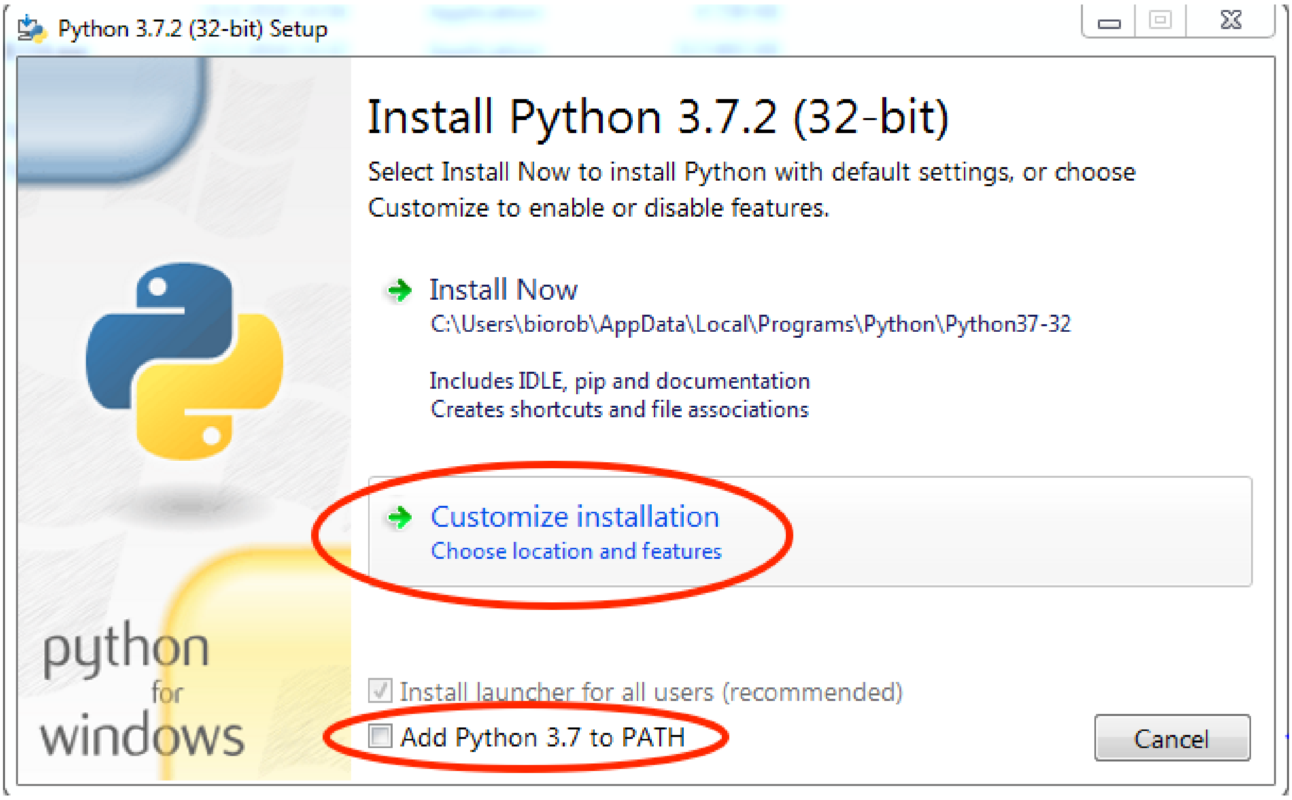
If either of them returns `Python 3` then you can skip the Python installation section and continue with the rest.

2.1.2 Installation

To download and install Python use the link : Python-Windows

During installation, when you see the pop up window figure 1 make sure you check on the box *Add Python 3.7 to PATH* and you click on the customize installation option.

Next, you will see figure 2 where you need to make sure all the check boxes are ticked. Finally in the advanced options step like figure 3 you need to tick the choices like shown in the figure unless you are sure you know the implications of your choices.



Install Python 3.7.2 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

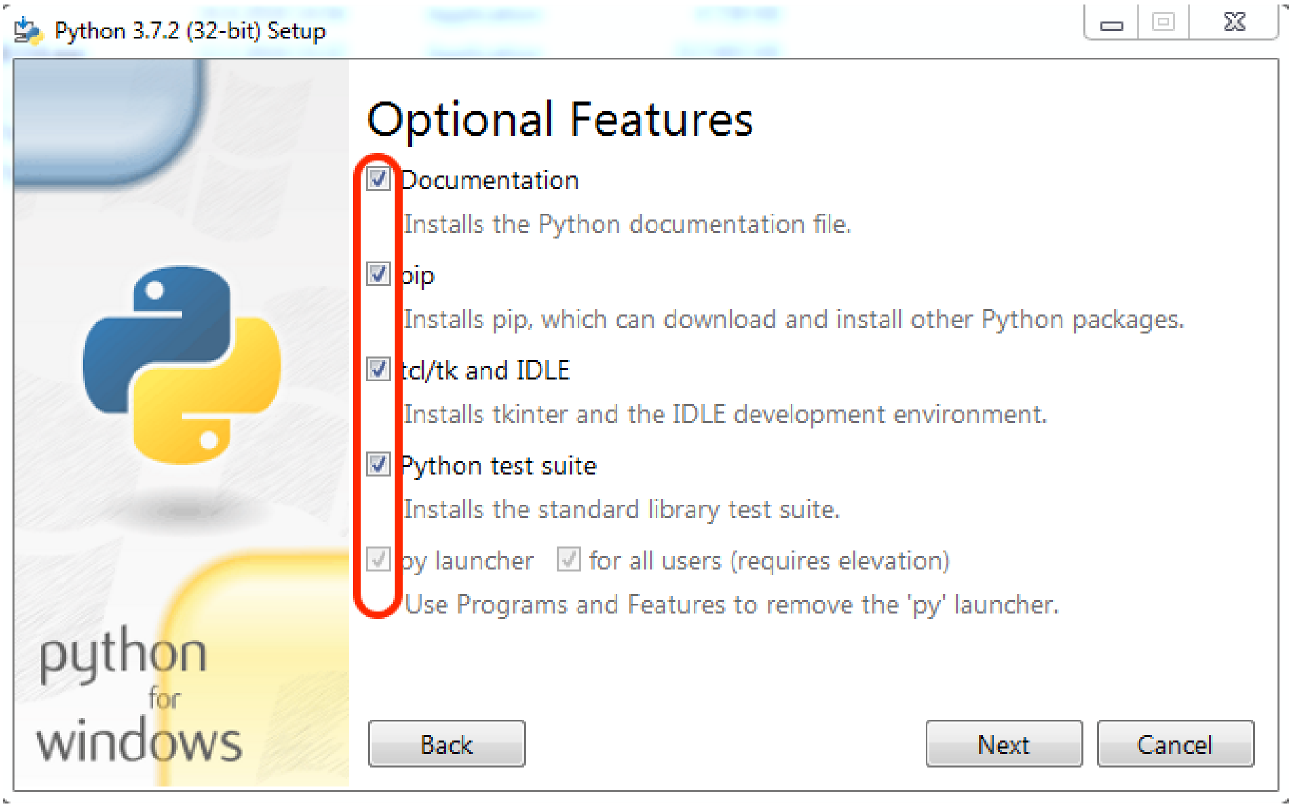
→ Install Now
C:\Users\biorob\AppData\Local\Programs\Python\Python37-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
□ Add Python 3.7 to PATH

Figure 1: Python installation customization - Step 1



Optional Features

☑ Documentation
Installs the Python documentation file.

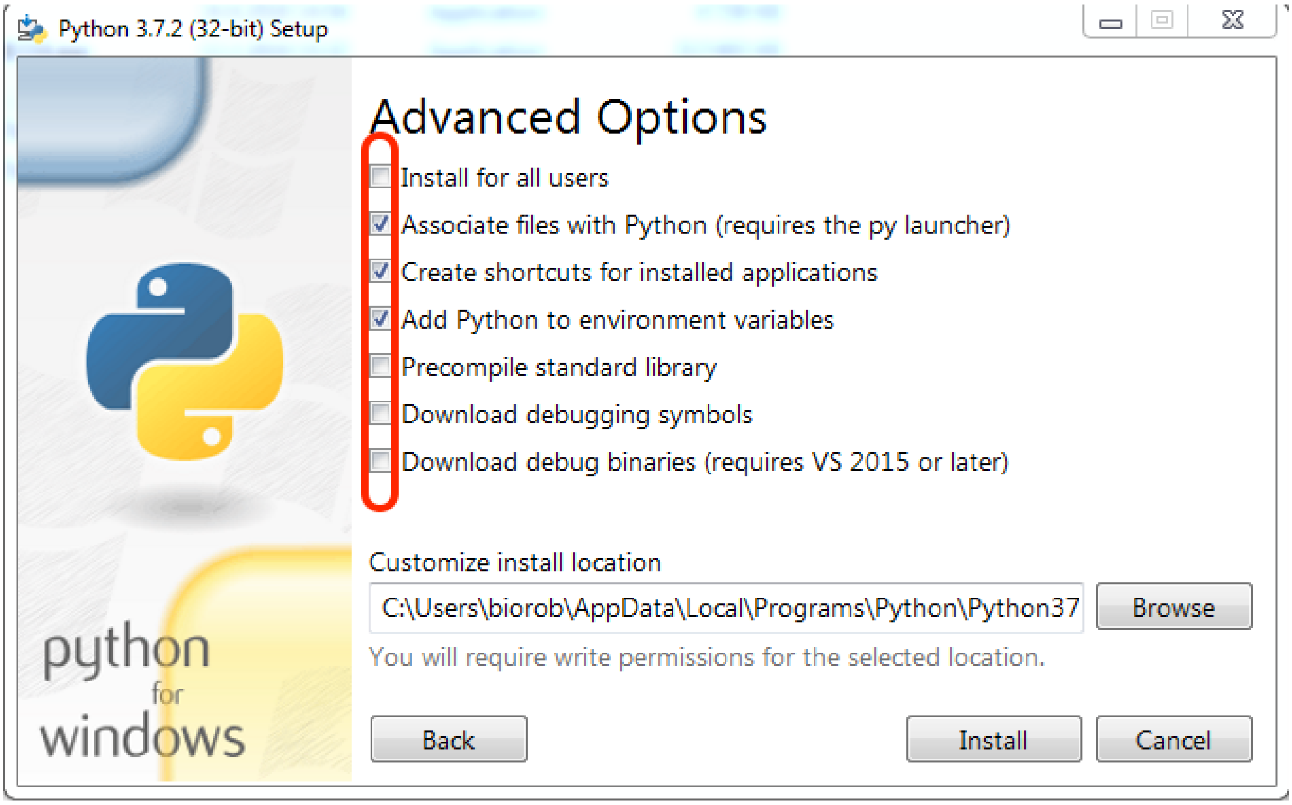
☑ pip
Installs pip, which can download and install other Python packages.

☑ tcl/tk and IDLE
Installs tkinter and the IDLE development environment.

☑ Python test suite
Installs the standard library test suite.

☑ py launcher ☑ for all users (requires elevation)
Use Programs and Features to remove the 'py' launcher.

Figure 2: Python installation customization - Step 2



Advanced Options

□ Install for all users
☑ Associate files with Python (requires the py launcher)
☑ Create shortcuts for installed applications
☑ Add Python to environment variables
□ Precompile standard library
□ Download debugging symbols
□ Download debug binaries (requires VS 2015 or later)

Customize install location

C:\Users\biorob\AppData\Local\Programs\Python\Python37

You will require write permissions for the selected location.

Figure 3: Python installation customization - Step 3

NOTE : *Install 3.7.2*

After installation to verify if everything is working open Git Bash again and run the above commands to check the python versions.

2.2 Pip

Python has a huge repository of packages that are widely used for different functions. In order to obtain these packages there are several package managers. The one we will be using during this course will be the official package installer for Python called *pip*.

2.2.1 Checking if pip exists

If you installed Python based on the instructions above then pip should be installed by default. Or it may have been already installed on your computer if Python had been pre-installed. To check if pip exists, open Git Bash and execute the following command:

NOTE : *pip or pip*3 *depends on your system. Typically they differentiate ones installed with python2 and python3 respectively.*

```
$ pip --version
```

```
$ pip3 --version
```

If *pip* is already installed then at least one of the above commands should print the version of the pip along with the python and its version associated with it. **Make sure that the python version is 3**

2.2.2 Installation

If you have verified that pip is not installed on your computer then in order to install pip you are expected to have either cloned or downloaded the exercise repository by now.

- Open Git Bash
- Navigate to the location where you have downloaded the exercise repository. You can use the command *cd* to change directories and *pwd* to check you current directory.
- Inside the exercise repository navigate to ***extras*** folder and execute the following command:

```
$ python get-pip.py
```

NOTE : *Make sure command **python** refers to python-3. To check use the commands mentioned in Python Installation section to get the corresponding python version. Accordingly use either python or python3 commands*

Check if you have installed everything correctly by referring to 2.2.1.

2.3 Spyder

Python programs can be written and run in several ways, it can be simply done on a terminal by running *python* or *ipython*. While this method is limited for simple programs, larger programs will be written using a text-editor or an Integrated Development Environment (IDE). Though it is not necessary to have an IDE for programming in Python, having one will bring many features that are useful while starting new

2.3.1 Installation

- Open Git Bash
- Execute the following command:

  ```
  $ pip install pyqt5==5.9.2
  ```

  or

  ```
  $ pip3 install pyqt5==5.9.2
  ```

  Use of *pip* or *pip*3 depends on the one associated with python3. If you are not sure refer back to section 2.2.1
- Next, install spyder with the command:

  ```
  $ pip install spyder
  ```

  or

  ```
  $ pip3 install spyder
  ```

2.3.2 Checking spyder

To check if spyder is installed, execute the following command from Git Bash

```
$ spyder3
```

If everything is working then Spyder IDE should open and you are ready to begin with the exercises.

3 Mac-OSX

3.1 Python

3.1.1 Checking if python exists

Before installing please check if you already have python installed on your computer. To do so open terminal application. Once terminal is open execute the following commands,

```
$ python –V
```

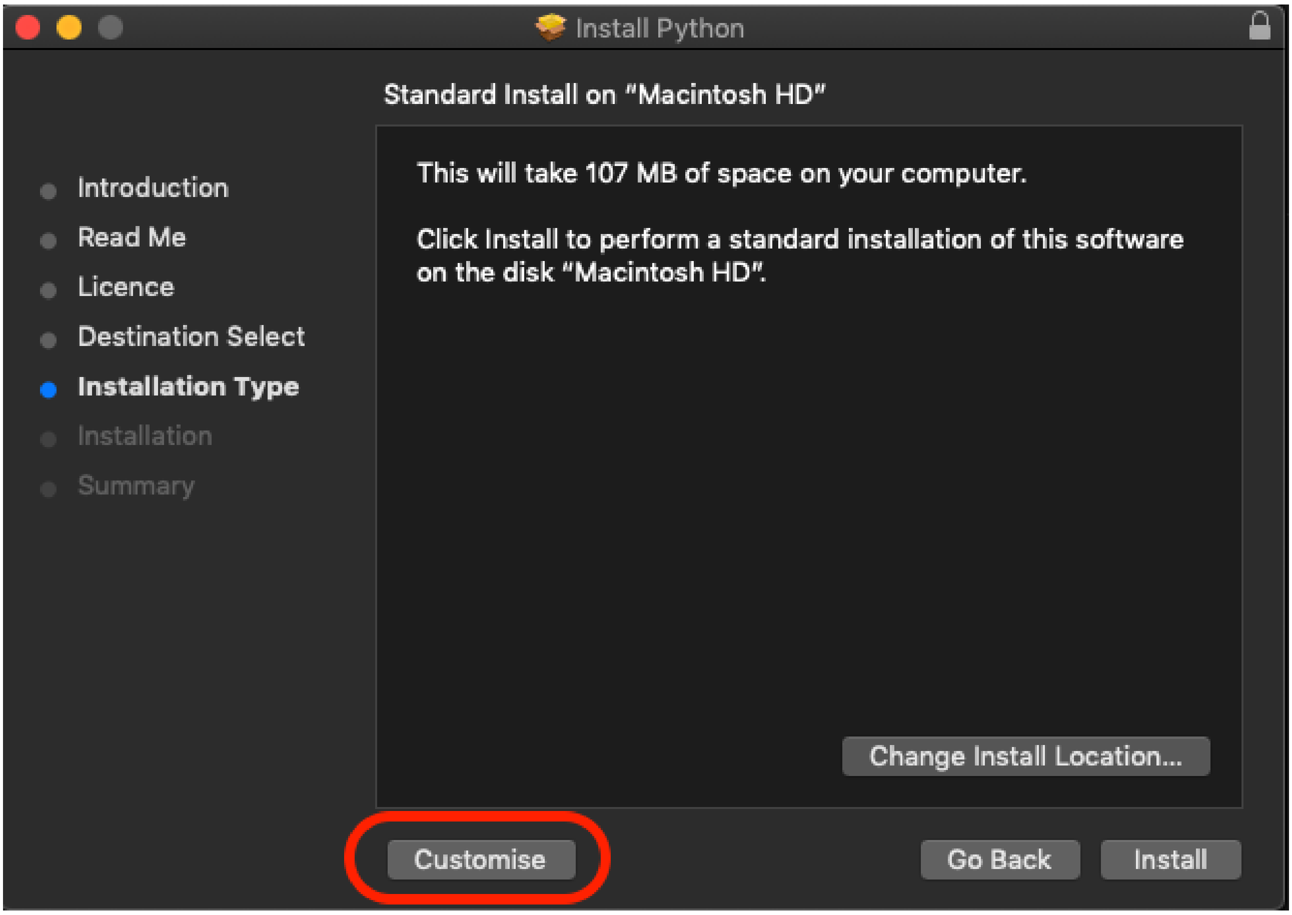
```
$ python3 –V
```

If either of them return `Python 3` then you can skip the Python installation section and continue with the rest.

3.1.2 Installation

To download and install Python use the link : MacOS

During installation step make sure you choose customize option like in figure 4 and then confirm that all the check boxes are selected like in figure 5

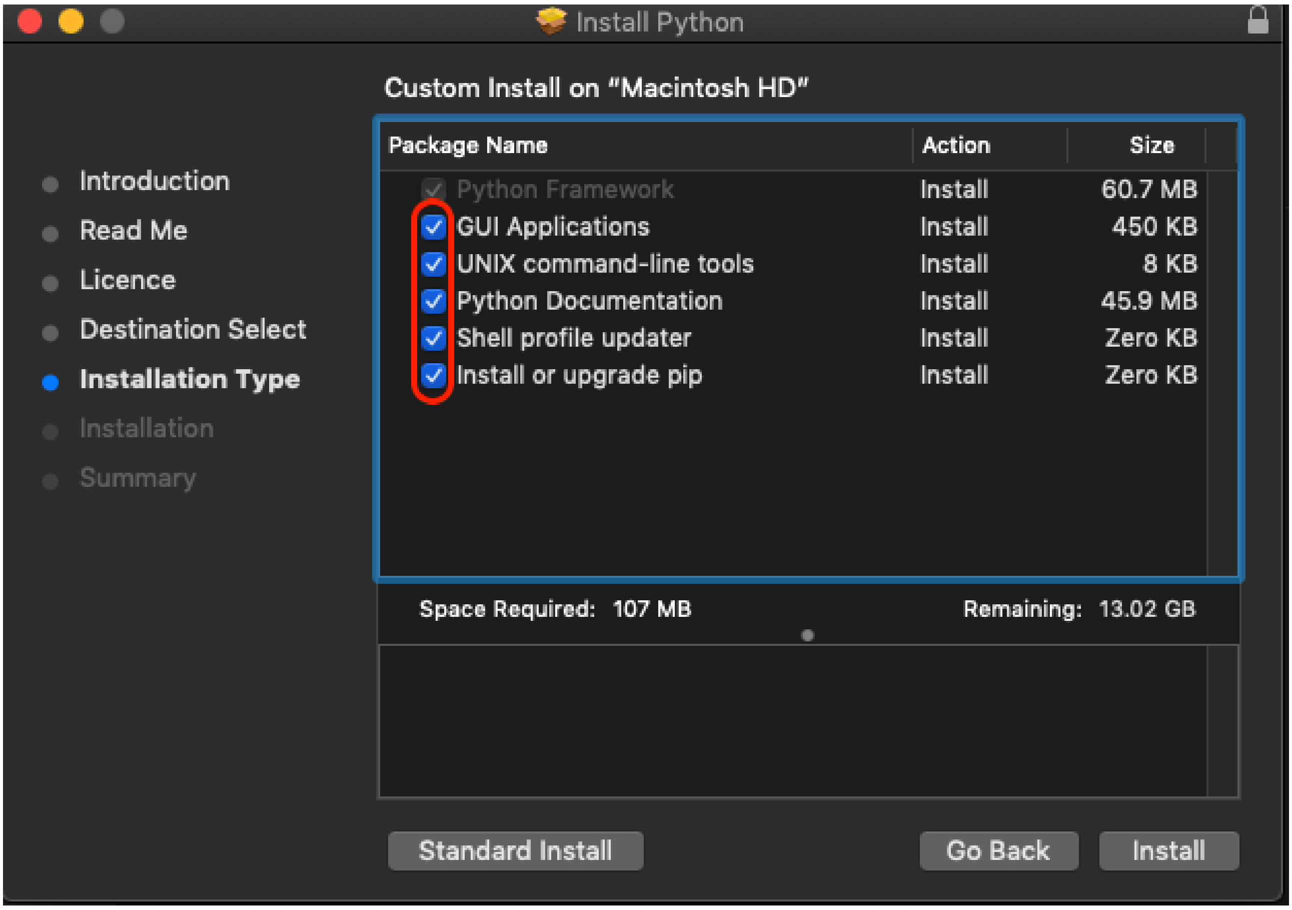


Standard Install on "Macintosh HD"

This will take 107 MB of space on your computer.

Click Install to perform a standard installation of this software on the disk "Macintosh HD".

Figure 4: Python installation customization - Step 1



Custom Install on "Macintosh HD"

Package Name	Action	Size
☑ Python Framework	Install	60.7 MB
☑ GUI Applications	Install	450 KB
☑ UNIX command-line tools	Install	8 KB
☑ Python Documentation	Install	45.9 MB
☑ Shell profile updater	Install	Zero KB
☑ Install or upgrade pip	Install	Zero KB

Space Required: 107 MB Remaining: 13.02 GB

Figure 5: Python installation customization - Step 2

NOTE : *Install 3.7.2*

After installation to verify if everything is working open terminal again and run the commands in section 3.1.1.

3.2 Pip

Python has a huge repository of packages that are widely used for different functions. In order to obtain these packages there are several package managers. The one we will be using during this course will be the official package installer for Python called *pip*.

3.2.1 Checking if pip exists

If you installed Python based on the instructions above then pip should be installed by default. Or it may have been already installed on your computer if Python had been pre-installed. To check if pip exists, open terminal and execute the following command:

NOTE : *pip or pip*3 *depends on your system. Typically they differentiate ones installed with python2 and python3 respectively.*

```
$ pip −−version
```

```
$ pip3 −−version
```

If *pip* is already installed then at least one of the above commands should print the version of the pip along with the python and its version associated with it. **Make sure that the python version is 3**

3.2.2 Installation

If you have verified that pip is not installed on your computer then in order to install pip you are expected to have either cloned or downloaded the exercise repository by now.

- Open terminal application
- Navigate to the location where you have downloaded the exercise repository. You can use the command *cd* to change directories and *pwd* to check you current directory.
- Inside the exercise repository navigate to ***extras*** folder and execute the following command:

```
$ python get-pip.py
```

NOTE : *Make sure command* ***python*** *refers to python-3. To check use the commands mentioned in Python Installation section to get the corresponding python version. According use either python or python3 commands*

Check if you have installed everything correctly by referring to 3.2.1.

3.3 Spyder

Python programs can be written and run in several ways, it can be simply done on a terminal by running *python* or *ipython*. While this method is limited for simple programs, larger programs will be written using a text-editor or an Integrated Development Environment (IDE). Though it is not necessary to have an IDE for programming in Python, having one will bring many features that are useful while starting new

3.3.1 Installation

- Open terminal
- Execute the following command:

  ```
  $ pip install pyqt5==5.9.2
  ```

  or

  ```
  $ pip3 install pyqt5==5.9.2
  ```

  Use of *pip* or *pip*3 depends on the one associated with python3. If you are not sure refer back to section 3.2.1
- Next, install spyder with the command:

  ```
  $ pip install spyder
  ```

  or

  ```
  $ pip3 install spyder
  ```

3.3.2 Checking spyder

To check if spyder is installed, execute the following command from a terminal.

```
$ spyder3
```

If everything is working then Spyder IDE should open and you are ready to begin with the exercises.

4 Linux

NOTE : *These instruction are for Ubuntu or other Debian-based distributions. The setup for other Linux distributions should be adapted accordingly.*

4.1 Python

4.1.1 Checking if python exists

Before installing please check if you already have python installed on your computer. To do so open terminal application. Once terminal is open execute the following commands,

```
$ python -V
```

```
$ python3 -V
```

If either of them return `Python 3` then you can skip the Python installation section and continue with the rest.

4.1.2 Installation

To download and install Python use the link :

- Open terminal application
- Execute the following command:

  ```
  $ sudo apt-get install python3
  ```

  The above command will ask you to enter your system password before beginning the installation process.

After installation to verify if everything is working open terminal again and run the commands in section 4.1.1.

4.2 Pip

Python has a huge repository of packages that are widely used for different functions. In order to obtain these packages there are several package managers. The one we will be using during this course will be the official package installer for Python called *pip*.

4.2.1 Checking if pip exists

If you installed Python based on the instructions above then pip should be installed by default. Or it may have been already installed on your computer if Python had been pre-installed. To check if pip exists, open terminal and execute the following command:

NOTE : *pip or pip3 depends on your system. Typically they differentiate ones installed with python2 and python3 respectively.*

```
$ pip --version
```

```
$ pip3 --version
```

If *pip* is already installed then at least one of the above commands should print the version of the pip along with the python and its version associated with it. **Make sure that the python version is** 3

4.2.2 Installation

If you have verified that pip is not installed on your computer then in order to install pip you are expected to have either cloned or downloaded the exercise repository by now.

- Open terminal application
- Execute the following command:

  ```
  $ sudo apt−get install python3−pip
  ```

Check if you have installed everything correctly by referring to 4.2.1.

4.3 Spyder

Python programs can be written and run in several ways, it can be simply done on a terminal by running *python* or *ipython*. While this method is limited for simple programs, larger programs will be written using a text-editor or an Integrated Development Environment (IDE). Though it is not necessary to have an IDE for programming in Python, having one will bring many features that are useful while starting new

4.3.1 Installation

- Open terminal
- Execute the following command:

  ```
  $ pip install pyqt5==5.9.2
  ```

  or

  ```
  $ pip3 install pyqt5==5.9.2
  ```

  Use of *pip* or *pip*3 depends on the one associated with python3. If you are not sure refer back to section 4.2.1
- Next, install spyder with the command:

  ```
  $ pip install spyder
  ```

  or

  ```
  $ pip3 install spyder
  ```

4.3.2 Checking spyder

To check if spyder is installed, execute the following command from a terminal

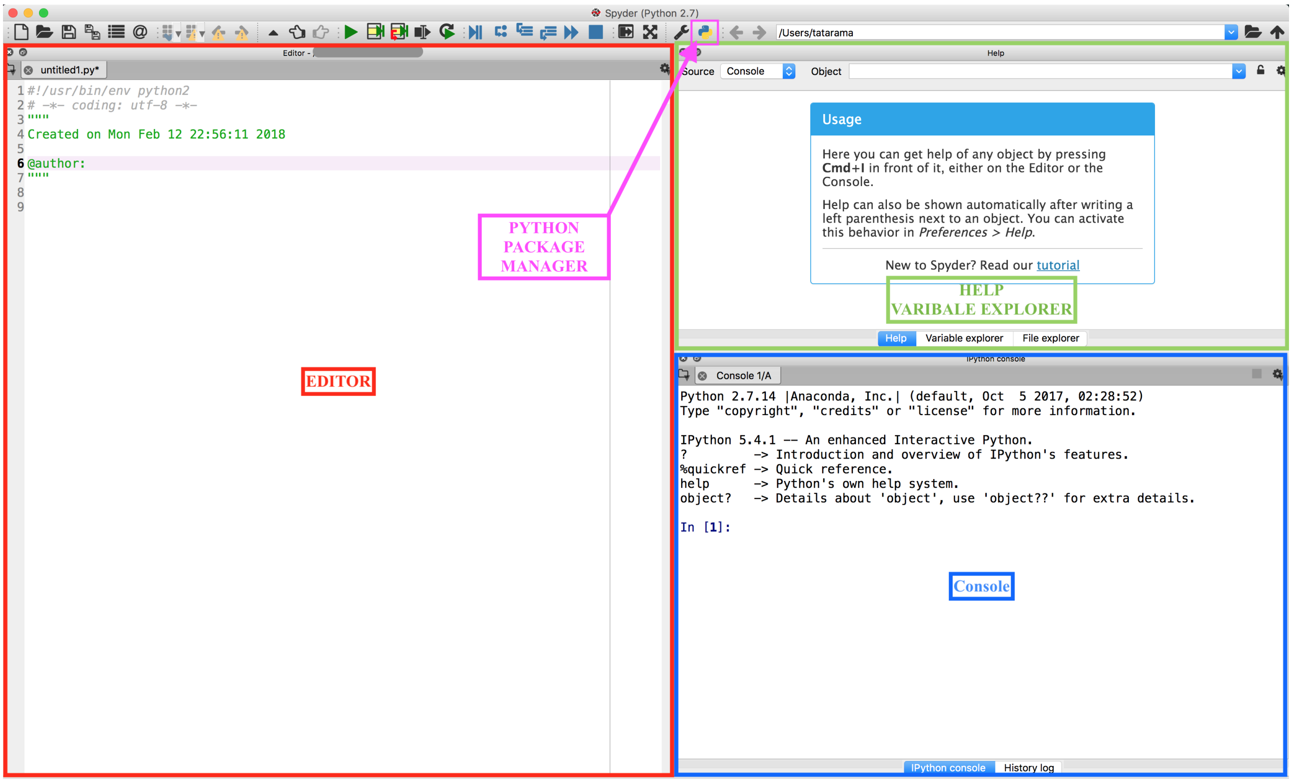
```
$ spyder3
```

If everything is working then Spyder IDE should open and you are ready to begin with the exercises.

5 Spyder

Spyder can be run by executing the command *spyder*.

Figure 6 shows the main windows when you first open Spyder. As you may have noticed, the layout is similar to Matlab. Spyder has very good documentation to get you started. It is recommended to go through the steps to get familiarized with the IDE.



```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 12 22:56:11 2018

@author:
"""
```

Usage

Here you can get help of any object by pressing **Cmd+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help.*

New to Spyder? Read our tutorial

```
Python 2.7.14 |Anaconda, Inc.| (default, Oct  5 2017, 02:28:52)
Type "copyright", "credits" or "license" for more information.

IPython 5.4.1 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:
```

Figure 6: Spyder IDE overview

The three essential parts of the screen are outlined.

- **The console** (bottom right, marked in blue). You can work interactively here. Code run, either interactively or from the editor, will output any results here. Error messages will be reported here. There are two types of console: a Python console, and an IPython console. Both will run Python code, but we recommend the IPython console as it offers better visuals for debugging and has additional features.
- **The editor** (left, marked in red). You can write code to be saved to file or run here. This will suggest problems with syntax and has features to help debug and give additional information.
- **The inspector** (top right, marked in green). The Object inspector can display detailed help on specific objects (or functions, or...), and the Variable inspector can display detailed information on the variables that are currently defined. Extremely useful when debugging.
- **Python Package Manager** (top toolbar, marked in pink) Use this menu to add new paths to the default python package paths. (NOTE : The symbol may look slightly different on your machine from the one shown in this report)

For an in depth tutorial of Spyder follow the link.

5.1 Getting Help

In any python console you can get information about a particular python object using the *help* method. For example to get help on the type float,

```
help(float)
```

In spyder you can use this method in the console window. The Spyder environment also provides a panel in the top right corner (by default) which is the Object inspector. If you type float into the empty line in the Object inspector window, it will also provide the help string.

5.2 Debugging

Refer to the link debug

6 Programming with Python

6.1 Package installation

The final step before starting of with the exercise is to install a few necessary packages. We will be using pip to this.

- Open terminal (Git Bash on Windows)
- Navigate in the terminal to the exercise repository on your computer
- Execute the following command once you are in the root of the repository:

  ```
  $ pip install -r requirements.txt
  ```

  or

  ```
  $ pip3 install -r requirements.txt
  ```

  *Use pip or pip*3 *depending on the one that refers to python3.*

The *requirements.txt* installs the following packages:

- numpy : Scientific computing package for python
- matplotlib : Matlab like plotting tool for python
- cmc_pylog : Module for logging written for cmc

After successfully completing the installation steps in the previous sections, you can now get started with programming using Python. Python is not just a computational tool but a very powerful programming language. This means having to learn a few more extra concepts to get your job done. There are a ton of references available online for those who are interested in learning Python in depth. We will try to provide the necessary references to help with the concepts that are useful during the course as and when needed.

6.2 Basic Python Concepts

In this section we will quickly go over the list of topics given below. You can open and run the individual files marked with the same topic name using Spyder. We suggest you to go through each section individually and spend time exploring each of the concepts by making changes to the code and observing the outputs.

0. HelloWorld
1. Imports
2. Data Types
3. Math
4. Conditional Statements
5. Data Containers : Lists, Tuples and Dictionaries
6. Functions
7. Loops
8. Numpy
9. Matplotlib

10. Classes

While you are executing each of the small exercises, try to learn how to use different features of Spyder. Especially the help and debugging feature. When you are unsure of any command, use the help service either the one built into Python or Spyder. After familiarizing yourself with the above concepts try to solve the following python exercises.

6.3 Exercise 1

Check if the following matrix M is a magic square or not?

Hint : *A magic square is a square matrix which contains distinct integers and whose sum along any of its individual rows or columns or diagonal is a constant. The constant is called as a magic constant or magic sum or magic square*

$$M = \begin{bmatrix} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{bmatrix}$$

Further Step : *Try if you can generalize your script to have a function to check any arbitrary matrix if it is a magic square or not. Import the function as a module in another script and use it to check the matrix M*

6.4 Exercise 2 - Plotting a function

Plot the following function $f(x)$ over an interval [0, 2] with proper labels and title

$$f(x) = sin(x-2)e^{-x^2}$$

7 References and additional useful links

7.1 Python

- NumPy for MATLAB users
- A byte of python
- A Crash Course in Python for Scientists
- The official Python documentation should be your first stop when looking for information
- numpy
- scipy
- matplotlib

7.2 Git

- Try Git!
- A Visual Git Reference

7.3 Spyder

- Getting started with Spyder