Lab 1

Yapi D

September 24, 2019

1 Objective

In this first lab you will install all tools needed to program in python and learn about python function, function arguments, variable and learn about two pythons packages: Matplotlib for plotting and Pandas for reading csv/excel files. I will explain the concept of python package as well.

2 Installing all requirements

The first step is to install all the requirements we need to program in Python. You start by installing Python, the text editor Atom. Then you create a Github account. Github is a cloud server that allows you to save your code safely in a cloud environment. This is how modern software development is done. It allows you to cooperate with other software developers, by safely saving and sharing your code.

Follow these instructions:

1. Click here to install Python: Downloads > Download Python 3.7.4. Make sure to select Add Python path, and follow the instructions.
2. Install the text editor atom from here
3. Get a github account from here
4. Now go into your github account and create a repository called yourname_python_course
5. Install git on window from here
6. Invite me as a collaborator by clicking : Settings > Collaborators, then add my github username yellowsimulator in the box and click Add collaborator

2.1 Important linux and Window Powershell commands

```
#create a folder called myfolder
$ mkdir myfolder
```

```
#create a a file called myfile.py
$ New-Item myfile.py
```

```
#delate the file called myfile.py
$ rm myfile.py
```

```
#go inside the folder called myfolder
$ cd myfolder
```

```
#see what is inside the folder called myfolder
$ ls myfolder
```

```
#go back
$ cd ..
```

2.2 Importants git commands (git status, git add, git commit, git push)

```
#create a local copy of a folder call somerepository
$ git clone somerepository
```

To save your code on git

```
#check all files that chaged
$ git status

#add file called filename
$ git add filename

#confirm that you added the file
$ git commit -m "add your message here"

#now save your file on the cloud
$ git push
```

3 Installing more python stuff

Now we will create a virtual environment. A virtual environment allows you to manages all your python code nicely. You will learn why it is important.

```
#create a virtual environment called km_python_env
$ python -m venv km_python_env
```

```
#activate the virtual environement
$ km_python_env\Scripts\activate
```

```
#install numpy, matplotlib, pandas
$ pip install numpy
$ pip install matplotlib
$ pip install pandas
```

3.1 Exercise

1. create a local copy of the github repository you created before on you computer (use git clone)
2. go inside that repository and create a folder called lab1 (use cd and mkdir)
3. now create the file python_lab1.py inside the folder lab1 (use New-Item)

4 Function, variable, arguments and reading files

4.1 Function, variable and arguments

```
#create a function in python
def my_function():
  """
  This is my cool function
  """
  print("I am awesome")
```

Functions allows you to group your code that executes one task at the time. A variable holds a value of something. The message at the beginning of the function between """ """ is called doctrings. Why do we need it?

```
#create a function with a variable holding a string
def print_message():
  """
  This is my cool function
  with a variable
  """
  my_variable = "I am awesome"
  print(my_variable)
```

```
#create a function with one argument
def print_message(my_variable):
  """
  This is my cool function
  with one argument
  """
  print(my_variable)

#Initialize variable and call the function
my_variable = "I am awesome"
print_message(my_variable)
```

```
#create a function with two arguments
def print_message(name, job):
  """
  This is my cool function
  with two arguments.
  """
  message = "My name is {}, I am a {}".format(name,job)
  print(message)

#Initialize variables and call the function
name = "Yapi Thor "
job = " Musician"
print_message(name, job)
```

```
#create a function with many arguments
def print_arguments(**args):
"""
This function takes as many arguments
as you want
"""
print(args)

#Initialize variables and call the function
name1 = "bla"
name2 = "blabla"
print_arguments(name1, name2)
```

4.2 Importing python packages and reading a csv/exl file

Here you are introduced to the python package called pandas for reading data from a csv file or excel file and matplotlib for plotting the data. To read data from a csv or excel file use the function read_csv() and read_excel()

```
#A function that plots a freqeuncy spectrum
import pandas as pd
import matplotlib as plt

```

```
5
6  def plot_frequency_spectrum():
7    """
8    plot a frequency spectrum
9    """
10   path = "data.csv"
11   df = pd.read_csv(path)
12   frequency = df["frequency"].values
13   amplitude = df["amplitude"].values
14   plt.plot(frequency, amplitude)
15   plt.xlabel("Frequency")
16   plt.ylabel("Amplitude")
17   plt.title("Frequency spectrum")
18   plt.show()
19
20 # plot_frequency_spectrum()
```

4.2.1 Exercise

This exercise will introduce the if, else statement

1. Update the above function to take the path, x label, y label and title as arguments.
2. Now write a function that can plot data either from a csv (data.csv) file or an excel file data.xls.

```
1   #some hint
2   def my_function(path, ...):
3     """
4     This is my cool function
5     """
6     if  path == "data.csv":
7       df = pd.read_csv(path)
8     else:
9       df = pd.read_excel(path)
10
11
```