Setting Up a Devellopment Environment for Programming in Python

Auxilliary Material for the Lecture in Summer Term 2021

Stand February 15, 2021

Universität Regensburg
Fakultät Physik

1 Setting Up a Devellopment Environment for Programming in Python in Windows

The commands we write in Python need to be *interpreted*, i. e. a program needs to be installed that translates human-readable commands into computer language. This tutorial will show you how to install a *Python interpreter* and test its functionality.

The language Python can be expanded by adding modules. This gives flexibility, but also requires some "background settings". We do not want to delve into too much detail here. For this reason I recommend installing the package *Anaconda*, a ready-made software package comprising the interpreter, an editor and a ready-configured settings-subsystem.

The Python interpreter can be invoked in different manners. A very comfortable way of doing so is by means of the IDE [1] *Spyder*. This program is automatically installed by following the steps explained below.

It is still possible to work *without an IDE* – for this, see 1.3.

Read and follow each step exactly as described.
Yes, the entire document.

Deviating from the steps shown here can lead to an only partially functional installation. The tutors in this course are prepared for a working environment as described below. If you want to use different systems, full support cannot be guaranteed.

1.1 Downloadint the Installer

You'll find the intsallation package under `https://www.anaconda.com/products/individual`. Pick the current installer for Windows[2]. If you don't know which installer is the correct one for your system, chances are you should download the *64-Bit Graphical Installer (466 MB)* (cf. figure 1.1).

The file should be stored into your downloads folder. Locate the file

`Anaconda3-2020.11-Windows-x86_64.exe`

and execute it. (The file name may be slightly different, depending on when the devellopers of Anaconda publish a new version. Also, the installer for Mac obviously doesn't mention Windows. In any case, the file name should start with `Anaconda3` and end in `-x86_64`.) Cf. figure 1.2.

[1] *Integrated Development Environment* – a Code editor that has a direct connection to the interpreter
[2] Mac User obviously pick the latest installer for their operating system. I could not test this myself, but as far as I know, everything is analogous to the steps explained here.

Figure 1.1: Download Page of the Anaconda Installer

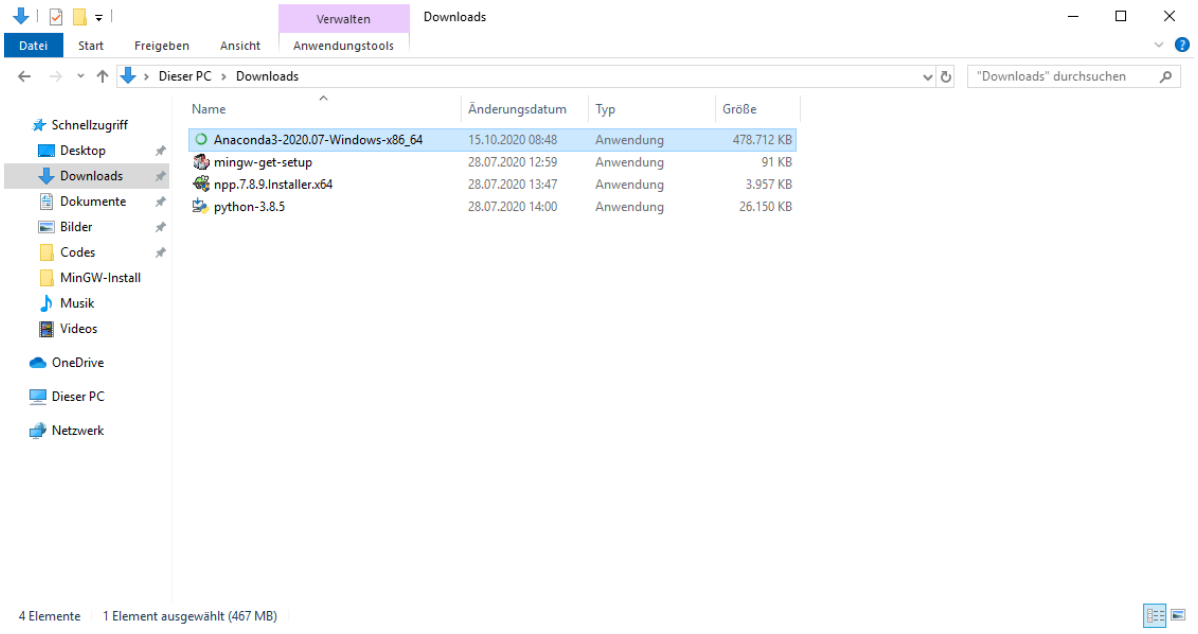

Figure 1.2: Anaconda Installer in the Downloads Directory

1.2 The Installation Itself

When launching the installer, you should see a welcome dialogue like in figure 1.3. Click *Next*. Agree to the Licence Agreement (figure 1.4). In the third dialogue window, select the option *Just for me (recommended)* (figure 1.5)

The next dialogue asks for the folder in which you want to install *Anaconda*. Whitespaces in path names often lead to problems. Thus I recommend selecting the path `C:\Anaconda\`. You'll need to create the subfolder `Anaconda` first. To do so, in the dialog in figure 1.6, click *Browse*. In the following dialog (figure 1.7) select *Local Disc (C:)*, click on *Make New Folder* and name it `Anaconda`. Select the newly created folder and click *OK*. You should now have returned to the original dialogue and see your created path in the text box (figure 1.8). Confirm this by clicking *Next*.

The next dialogue (Advanced Install Options, figure 1.9) can be left in the standard settings. Clicking *Next* starts the actuall installation (figure 1.10). It should finish within a couple of minutes.

After completing the installation, the installer will notify you about the online offers of the Anaconda team (figure 1.12). You can simply ignore this by clicking *Next*. In the last step, the installer offers to create bookmarks to the online content (figure 1.13). Again ignore this, i. e. clear all check boxes and close the installer by clicking *Finish*.

1.3 Test 1: Starting the Python Shell

The Python Shell is a minimal environment and can be used as a first test. To see whether the installation was successfull, first start the *Anaconda Prompt*. for this, click the start button and enter the search term *Anaconda*. Among the results you should find *Anaconda Prompt* or *Anaconda Powershell Prompt* (cf. figure 1.14). You can start either of them for this test.

You will see a black terminal window, in which you can enter the usual Windows console commands[3]. Enter now

```
python
```

With this, the Python interpreter takes over the console. That means that the console commands no longer work; instead you can now enter Python commands. Type now

```
print("Hello World!")
```

As a "response" you should now see the line `Hello World!` on screen. Figure 1.15 shows how in- and output should look like. If you can reproduce this: congratulations! The Python interpreter is now correclty installed. Continue with section 1.4. Otherwise, please ask your tutor for advice.

[3] If you have no prior experience with terminal windows, you'll learn everything you need in class. For now, you can "blindly" follow the instructions

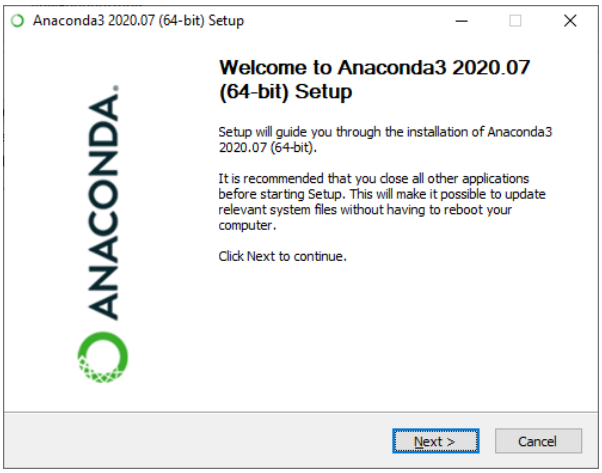

Figure 1.3: Anaconda Installer: Welcome Window

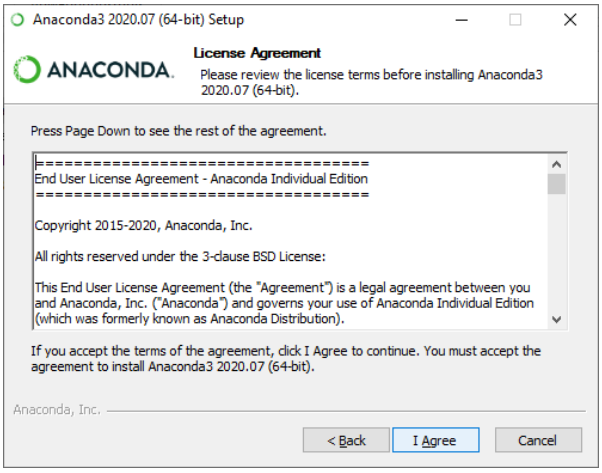

Figure 1.4: Anaconda Installer: Licence Agreement

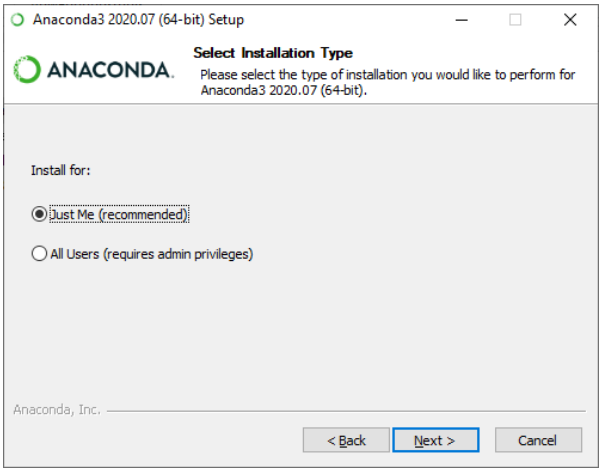

Figure 1.5: Anaconda Installer: User Group

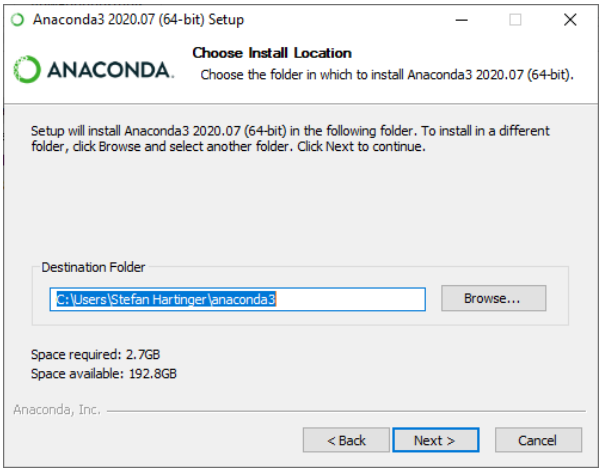

Figure 1.6: Anaconda Installer: Installation Path

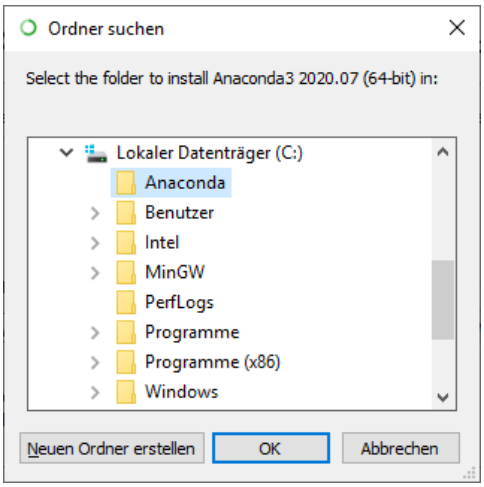

Figure 1.7: Anaconda Installer: Make New Folder

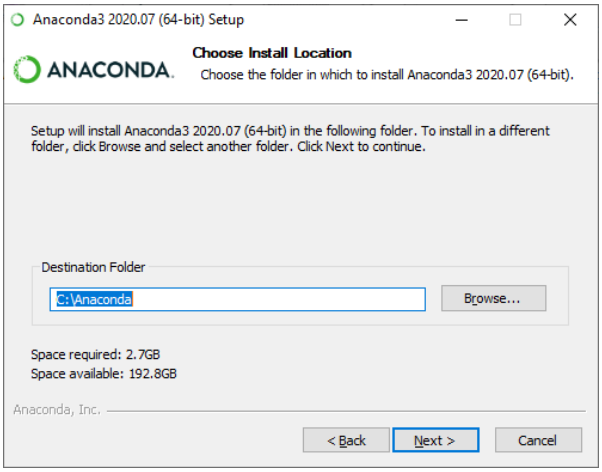

Figure 1.8: Anaconda Installer: Correct Installation Path

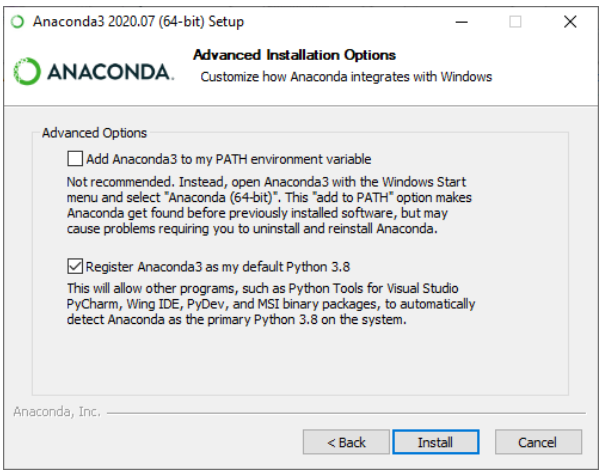

Figure 1.9: Anaconda Installer: Advanced Options

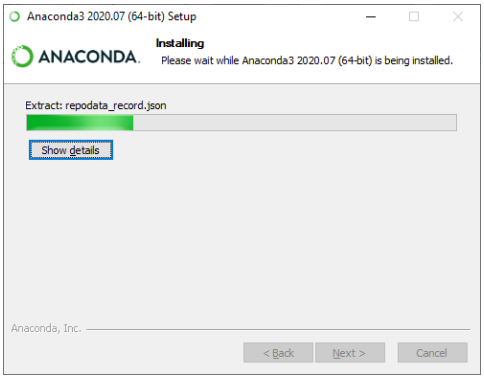

Figure 1.10: Anaconda Installer: Progress

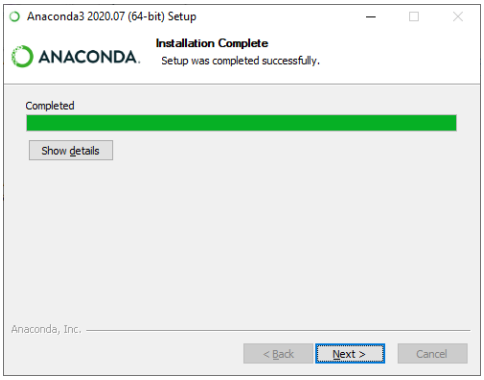

Figure 1.11: Anaconda Installer: Installation Complete

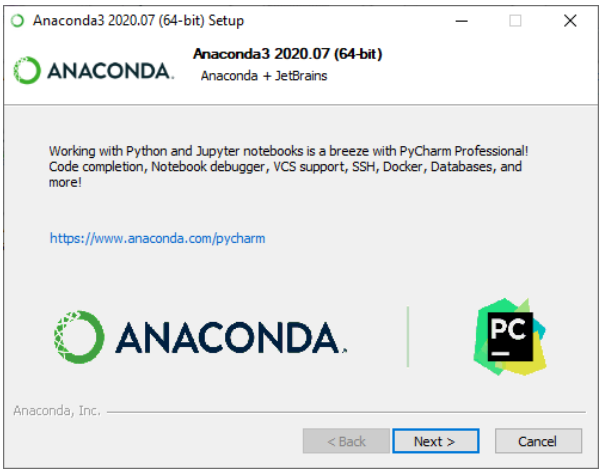

Figure 1.12: Anaconda Installer: Advertisement

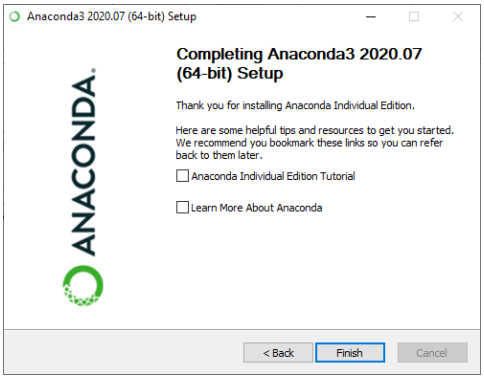

Figure 1.13: Anaconda Installer: Creating Bookmarks

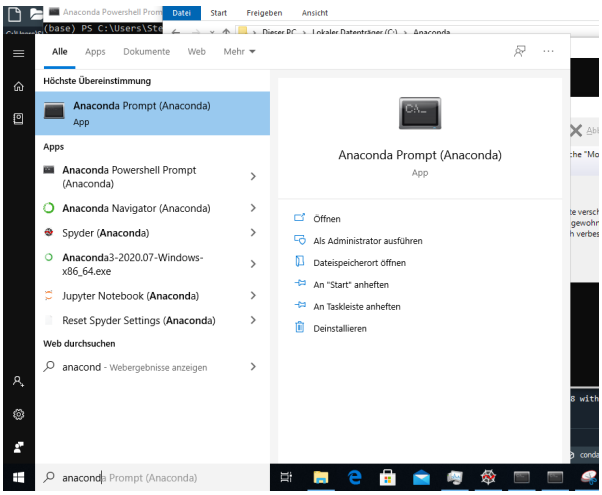

Figure 1.14: Starting the Anaconda Prompt

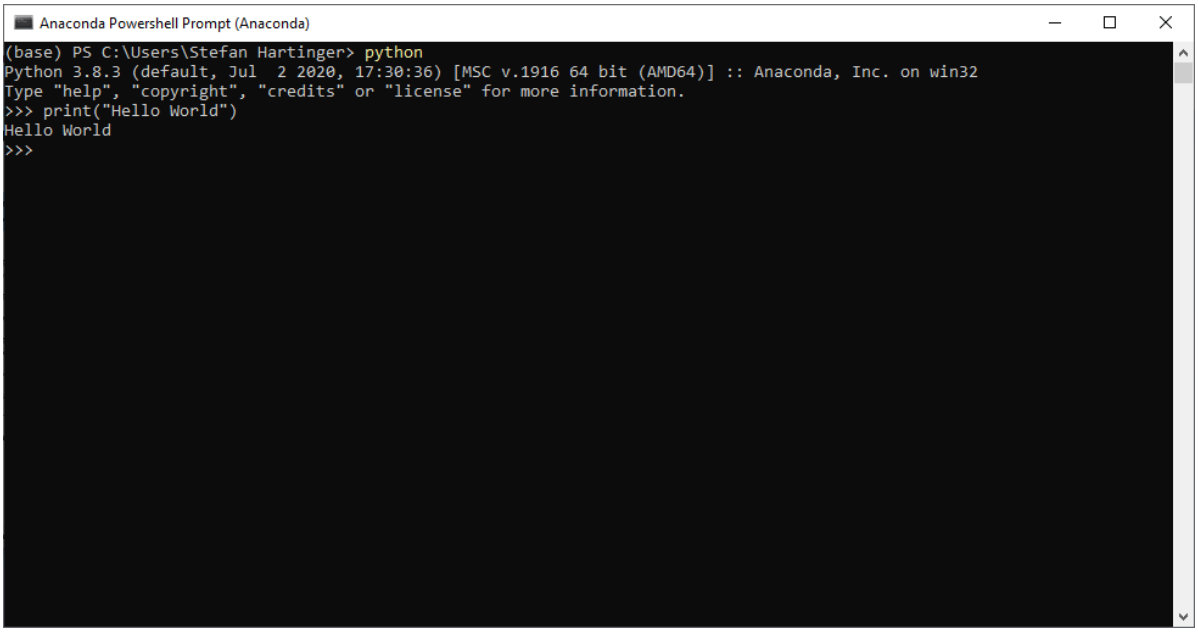

Figure 1.15: Hello World in the Interpreter Environment

1.4 Test 2: Spyder IDE

Now, start the program Spyder. For this, click Start, then enter the search term *Spyder* (figure 1.16). From the menu bar, select *File* → *Open* and load the file *Install-Test.py*, which you can download from GRIPS. Press F5 to execute the code. The result should look like figure 1.17. In particular, in the lower right part of the window you should now find the lines:

```
Interpreter up and running.
Loading numpy...okay.
Loading matplotlib...okay.
Creating a plot of sin(x)...okay
```

In the upper right part of the window, you'll see a button labelled *Plots*. Klick it to see the sine wave as shown in figure 1.18.

Did everything work as described here? Congratulations! You are now ready to commence the course. Write your codes in the code area (the part to the left of the window) and press F5 to execute them. You will find the output in the lower right part of the window. If there are any problems, don't hesitate to ask your tutors.

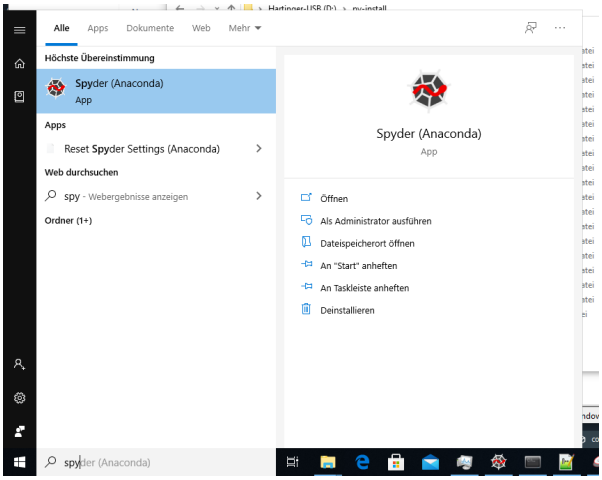

Figure 1.16: Launch Spyder IDE

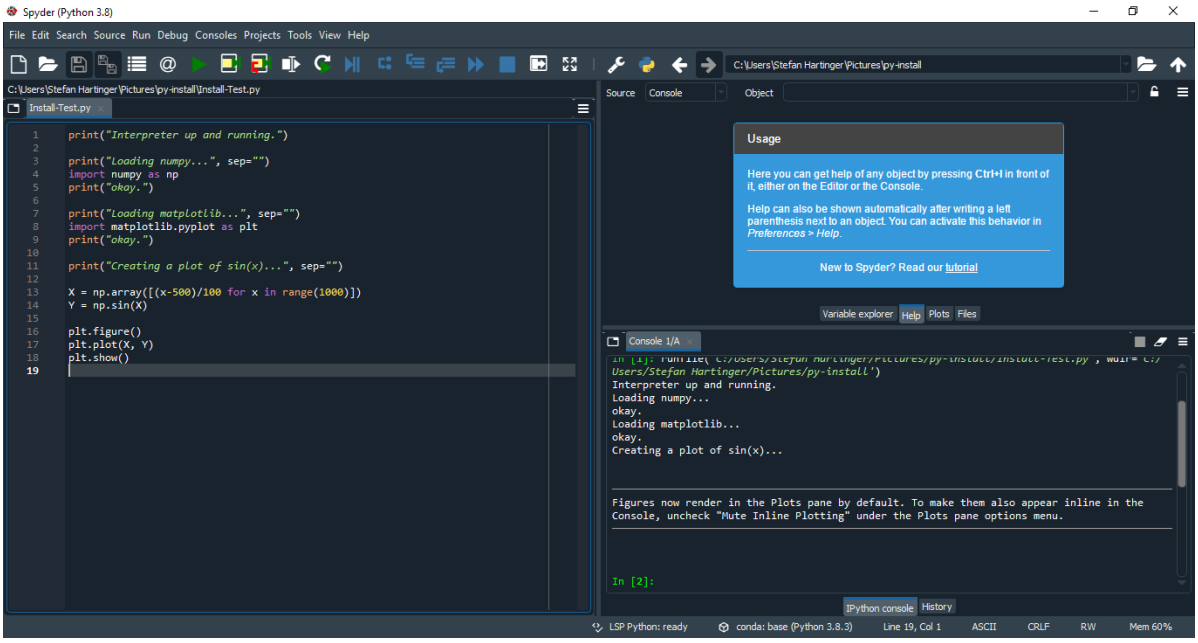

Figure 1.17: Spyder IDE

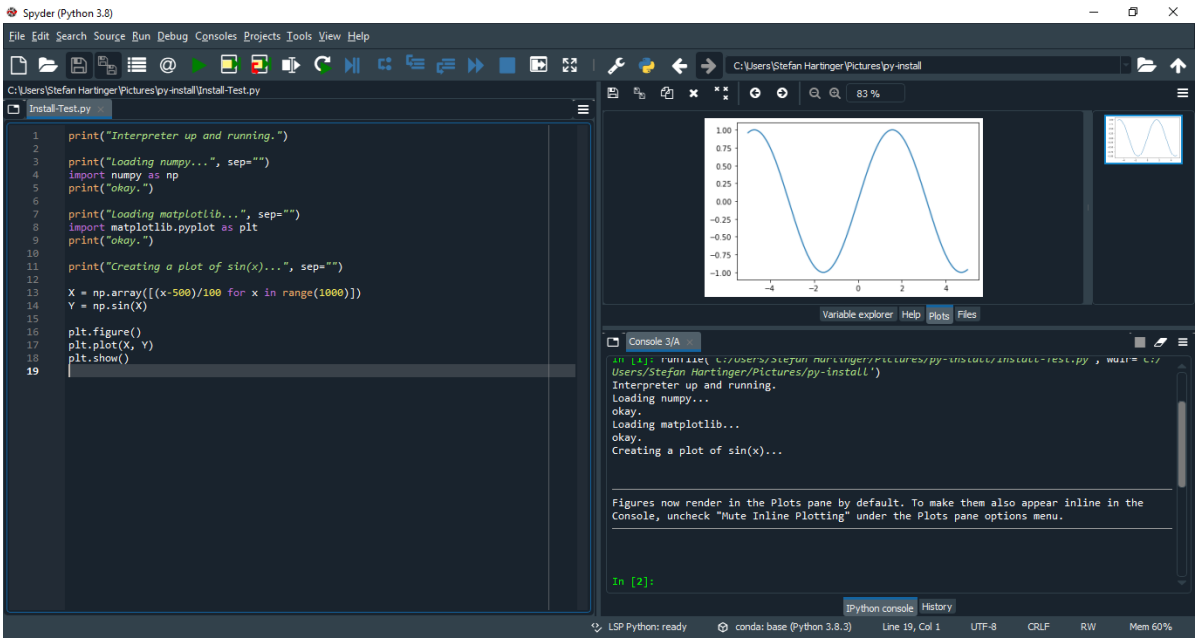

Figure 1.18: Spyder IDE: Plots

1.5 Optional: Lauching Scripts without IDE

This section assumes you have basic experience with console programs. Please ask your tutor if you need any assistance. You do not need to pass this section.

If you have any problems with Spyder or if you prefer working without an IDE, you can use any other code editor. I recommend notepad++ (can be installed for free from the Microsoft Store or downloaded from `https://notepad-plus-plus.org/downloads/`). Save your code as a `*.py` file. Launch the Anaconda Shell (cf. section 1.3) change the CWD to the directory where you saved your `*.py` files. For this, use the command `cd`.

Example: You've written the following code[4]:

[4] You'll learn in class what these lines do in detail.

Code

```
print("Interpreter up and running.")

print("Loading numpy...", end="")
import numpy as np
print("okay.")

print("Loading matplotlib...", end="")
import matplotlib.pyplot as plt
print("okay.")

print("Creating a plot of sin(x)...", end="")

X = np.array([(x-500)/100 for x in range(1000)])
Y = np.sin(X)

plt.figure()
plt.plot(X, Y)
plt.show()

print("okay.")
```

Store them as `C:\Codes\plotTest.py`. Start the Anaconda Shell and use `cd` to change the CWD to `C:\Codes`, i. e. type:

```
cd C:\Codes
```

Note that there is a whitespace between `cd` and `C:\Codes`.

Now you can execute your code by typing:

```
python plotTest.py
```

The Python interpreter will now run your code and return to the console thereafter.