

Guide to Anaconda

Table of contents

1	Introduction	1
2	Installing Anaconda	1
3	The Anaconda Prompt	2
4	Launching Jupyter Notebook	6
5	Environments with conda	8
5.1	Creating environments	9
5.2	Activating an environment	9
6	Workflow	9

1 Introduction

This is a short introduction to opening Jupyter Notebooks using the Anaconda Prompt. Jupyter Notebooks support interactive notebooks, which enable execution of specific parts of code and nice looking using Markdown, which is a quick and easy way of getting nicer looking text than plaintext. A cheatsheet can be found [here](#). Furthermore, notebooks are generally very popular in the community, so you will see them pop up all the time online.

2 Installing Anaconda

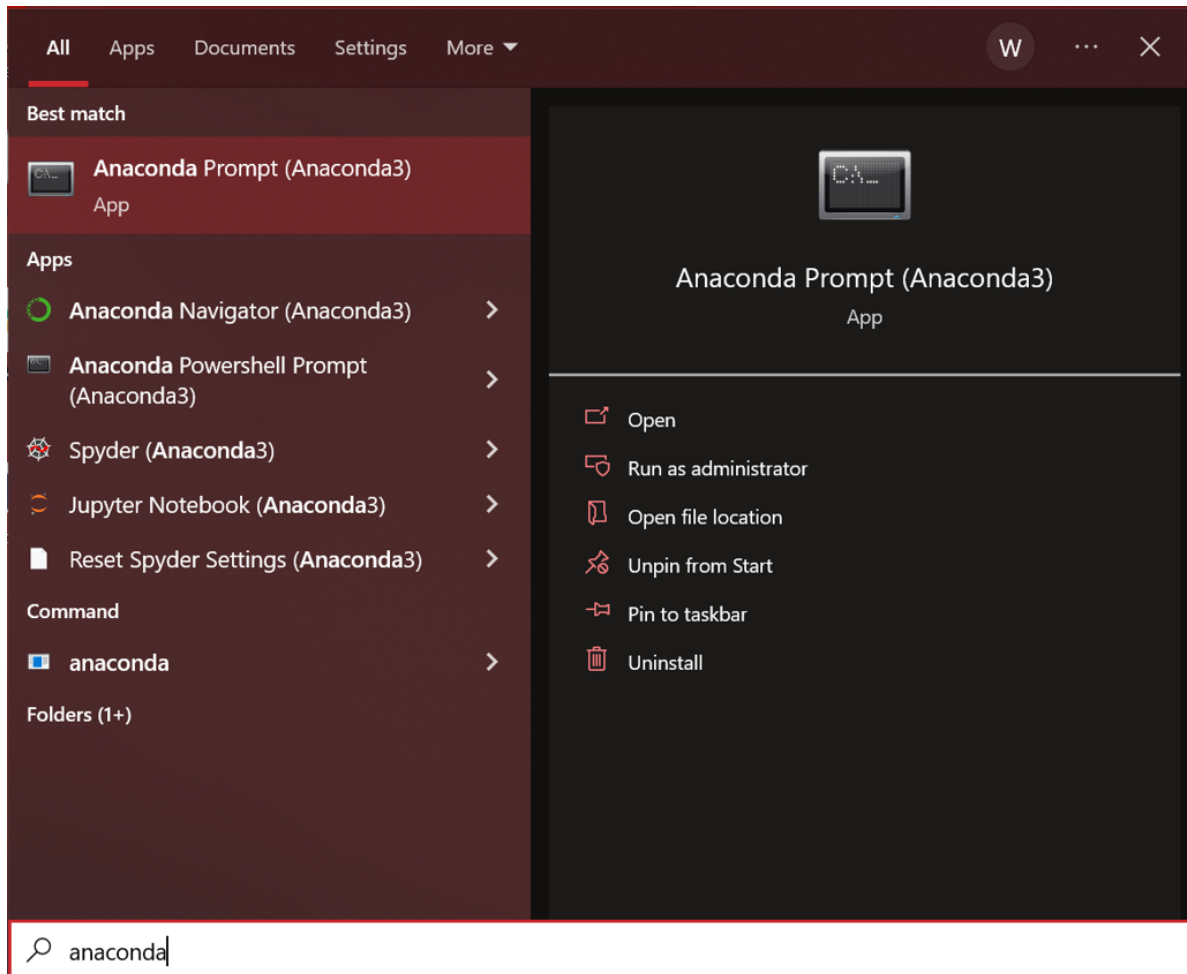
Anaconda can be downloaded through their [website](#). Later on you may also be able to download it through Statens IT, but for now we just use the base free version for students, academics and hobbyists – it's free, and it seems to cause no problems! When Anaconda has been

installed, you should now have an executable called *Anaconda Prompt* (possible *Anaconda Prompt (Anaconda3)*), which can be found through the search function.

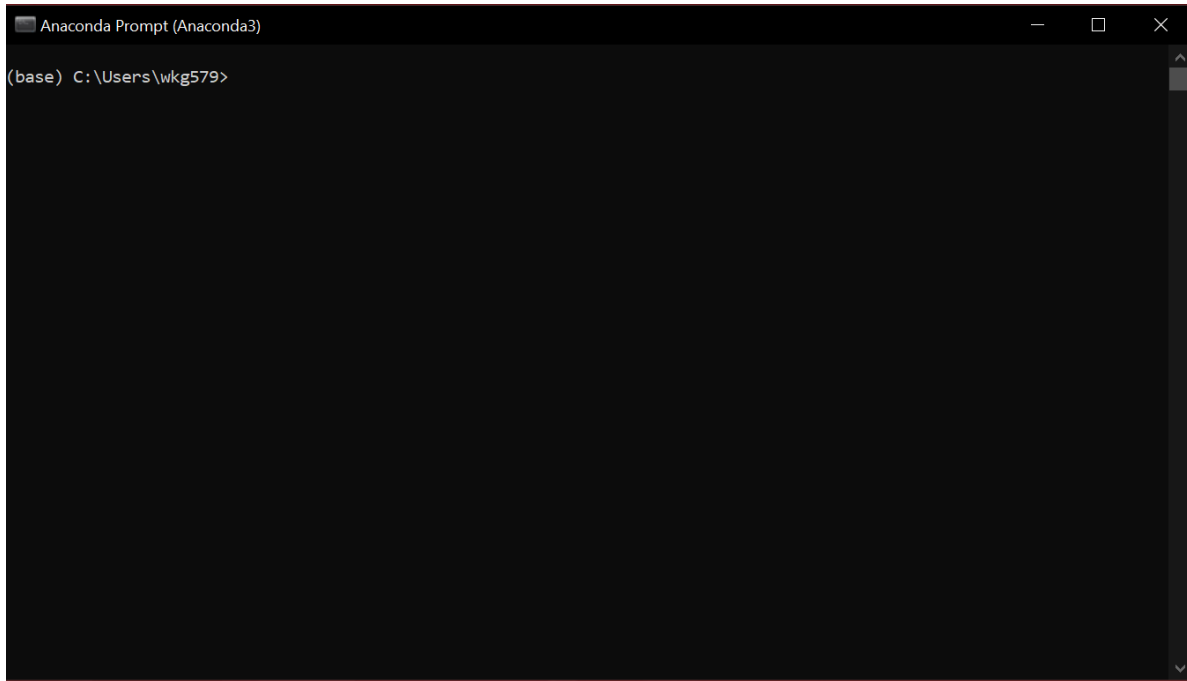
Note: Installing Anaconda may take a while.

3 The Anaconda Prompt

When searching for the Anaconda Prompt through the search function, it should look something like this:



Executing the program, you should get a nice grey box:

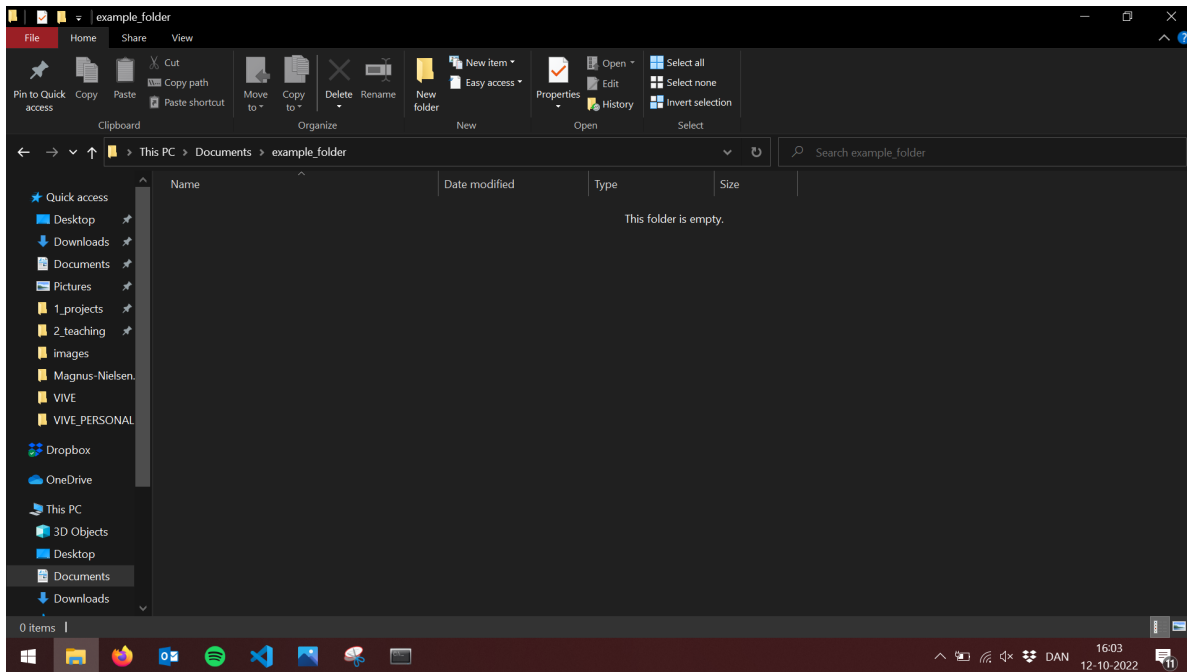


This is the Anaconda Prompt. This is from where we launch our Jupyter Notebook (or Jupyter Lab or any other IDE).

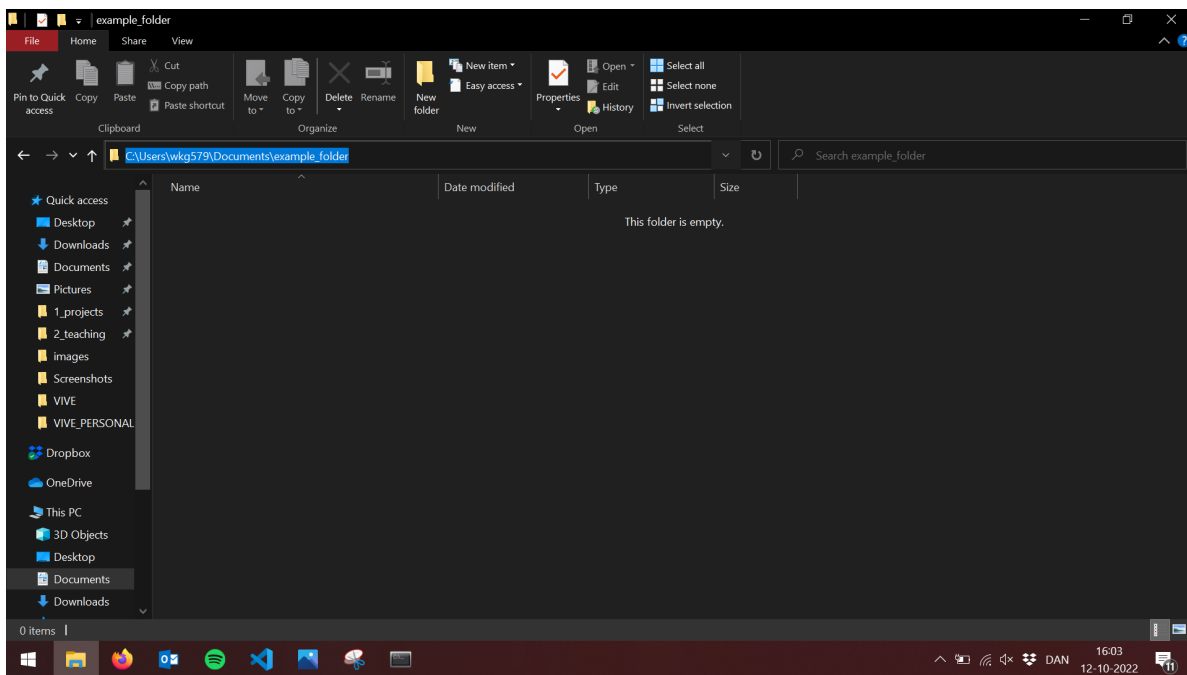
You should now navigate to the folder in which you have (or wish to have) your Jupyter Notebooks

For the purposes of this demonstration, I have created a folder in 'Documents' called 'example_folder' (Note: Spaces in folder and filenames are generally frowned upon as they can cause issues).

To find out the path of the folder, I open the folder in file explorer:



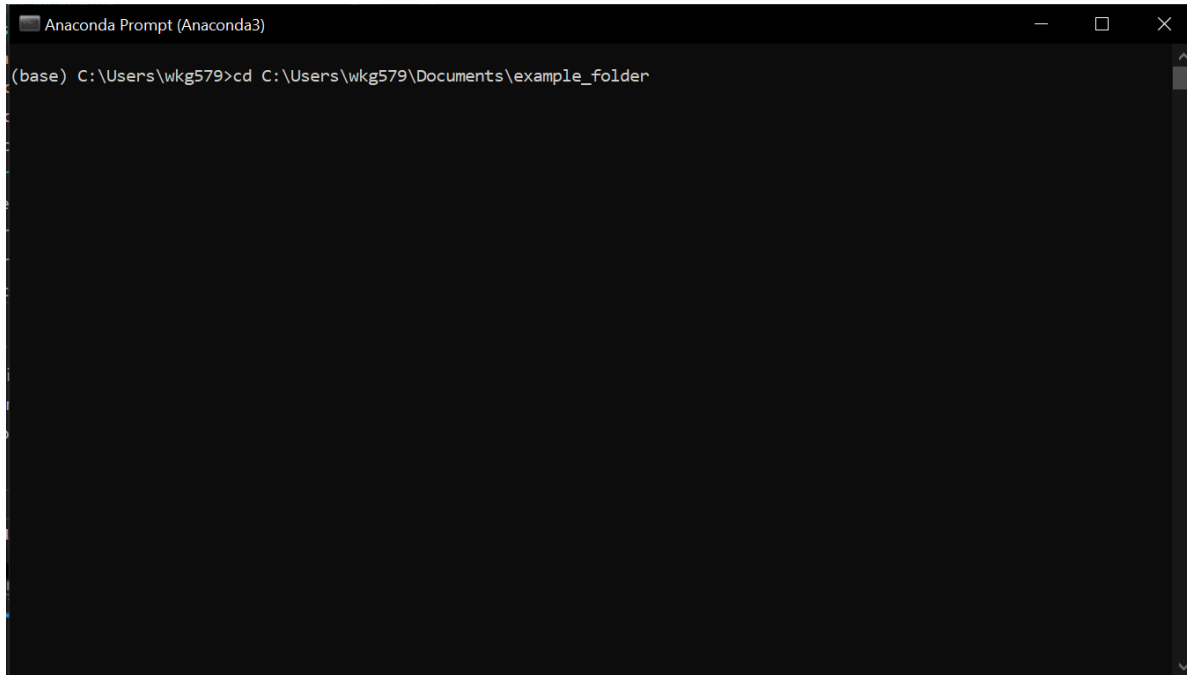
I then click at the top under the toolbar, which highlights the path for folder:



The path of this folder is `C:\Users\wkg579\Documents\example_folder`.

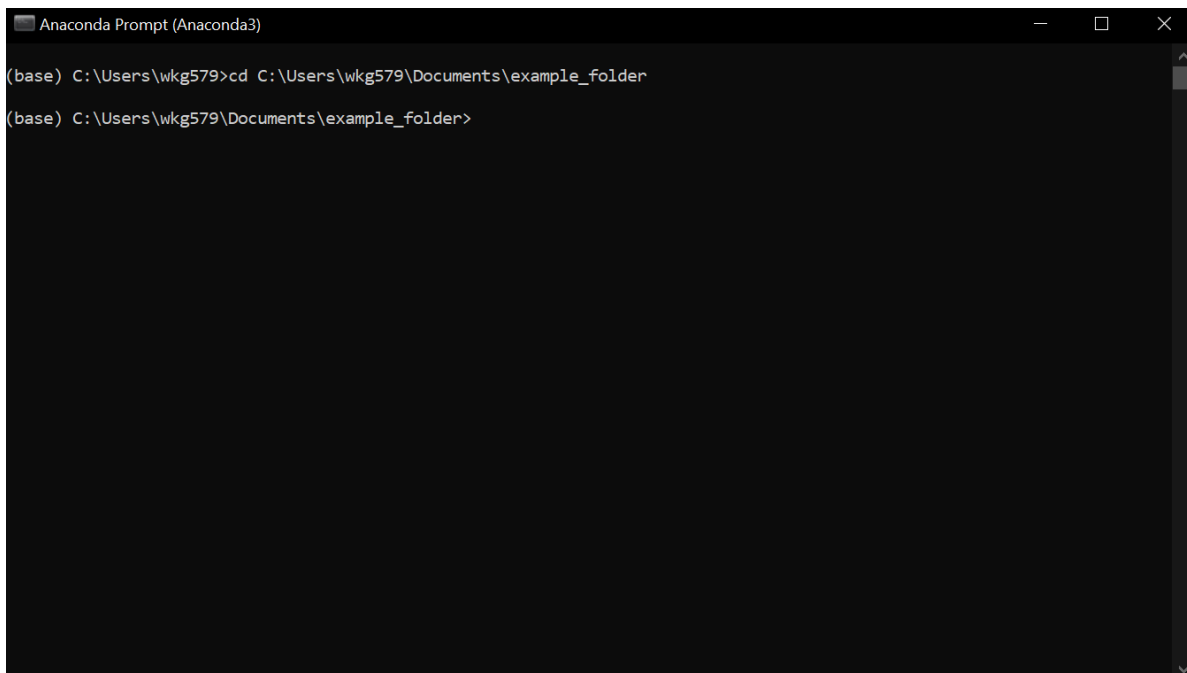
To change the folder (also called a directory) I use the commandline command `cd`, short for

change directory. The `cd` command consists of `cd path_you_want_to_go_to`. In my example, it would be `cd C:\Users\wkg579\Documents\example_folder`:



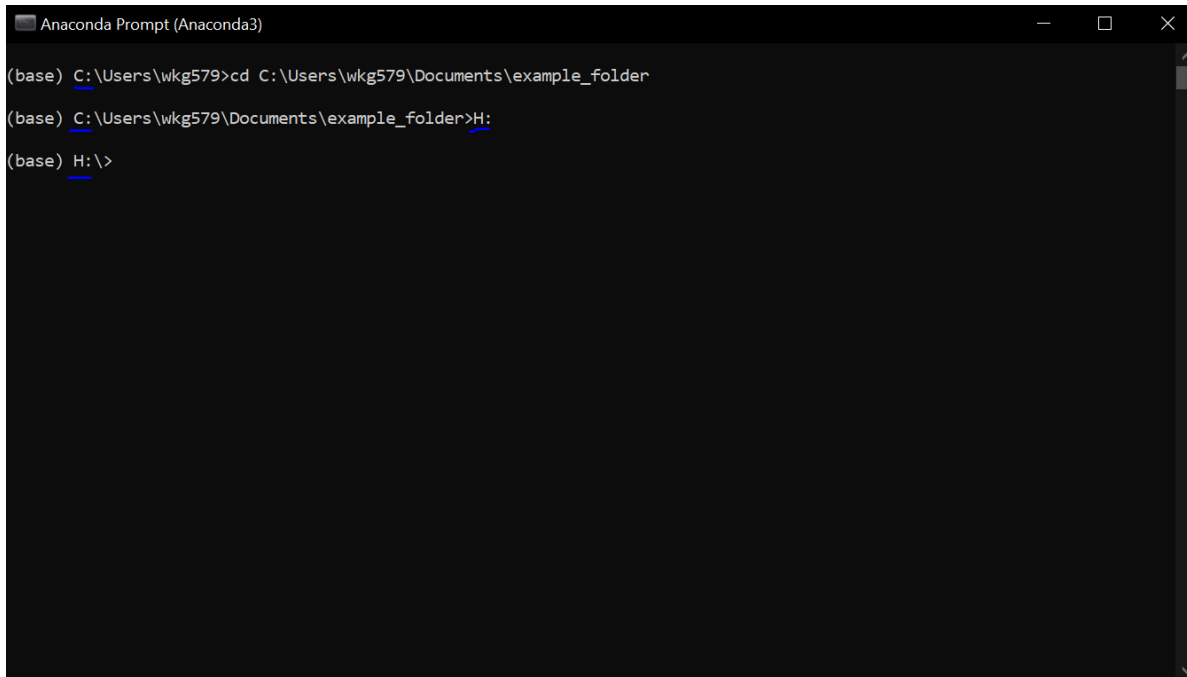
```
Anaconda Prompt (Anaconda3)
(base) C:\Users\wkg579>cd C:\Users\wkg579\Documents\example_folder
```

Which then changes the location from which further commands are launched:



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\wkg579>cd C:\Users\wkg579\Documents\example_folder
(base) C:\Users\wkg579\Documents\example_folder>
```

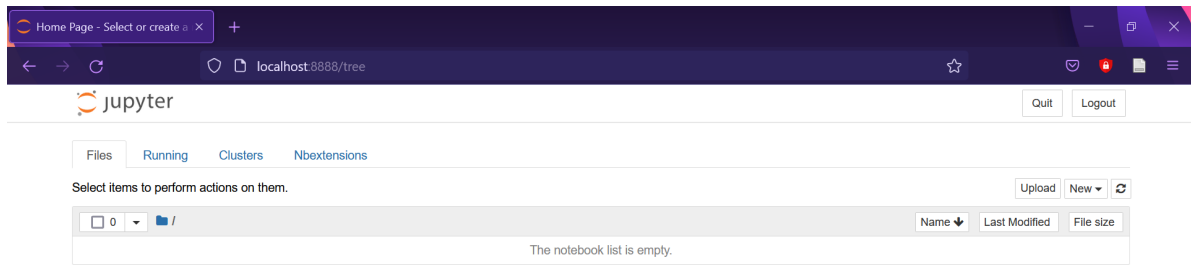
Note that my Anaconda Prompt and my project folder both are situated on the C-drive. If it so happens that your project folder is located on another drive, say the R-drive, you would need to also write the command `R:.` In my case I do not have an R-drive, but I have an H-drive:

A screenshot of an Anaconda Prompt window titled "Anaconda Prompt (Anaconda3)". The window has a dark background and a light gray title bar with standard Windows window controls. The command prompt shows the following sequence of commands and prompts: 1. The initial prompt is "(base) C:\Users\wkg579>". 2. The user enters "cd C:\Users\wkg579\Documents\example_folder", and the prompt changes to "(base) C:\Users\wkg579\Documents\example_folder>". 3. The user enters "H:", and the prompt changes to "(base) H:\>". 4. The user enters a backslash "\", and the prompt remains "(base) H:\>". The text is white on a dark background, with the prompt characters in a lighter shade.

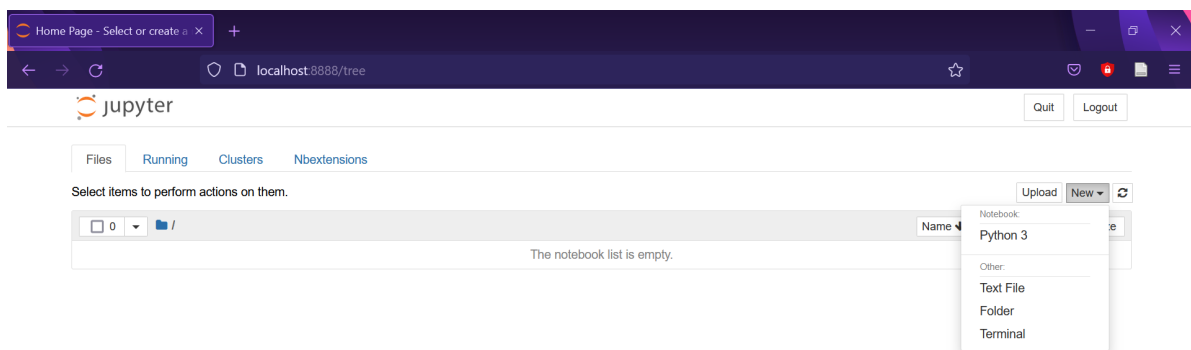
If you get an error message when trying to change drive (it cannot be found), make sure that your drive is accessible through the file explorer (sometimes it is not connected before you open it in the file explorer if you use a VPN).

4 Launching Jupyter Notebook

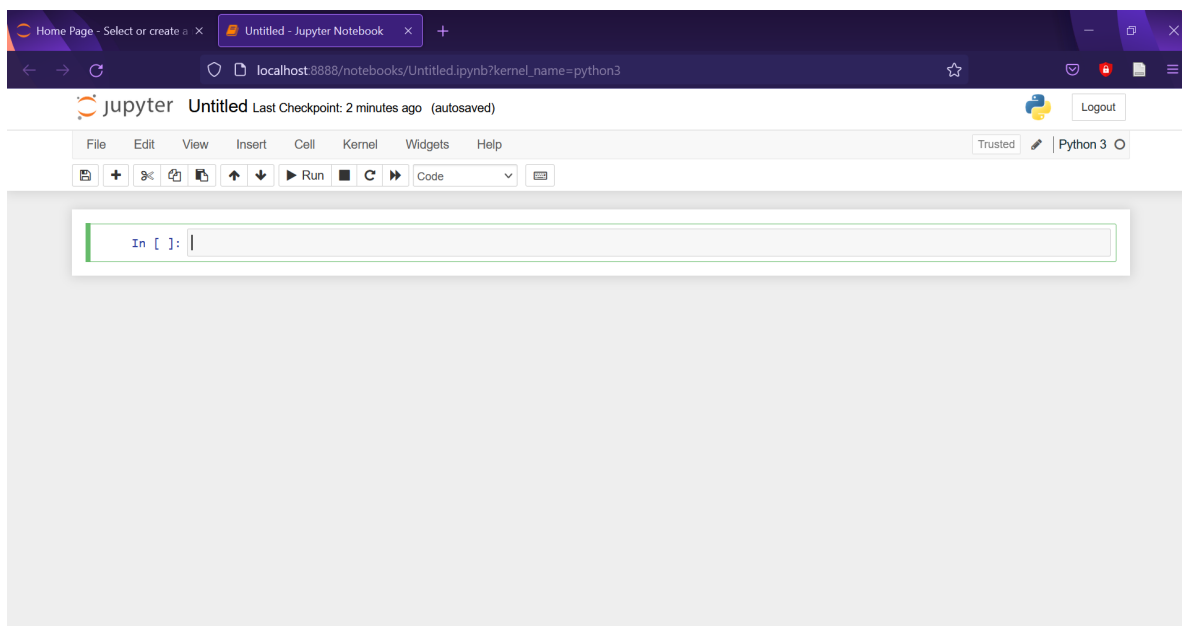
You are now located in your project folder, and can launch the Jupyter Notebook using the command `jupyter notebook`, which will open a browser where you can see your files:



As you can see, the folder is very empty, but you can easily create notebooks and folders using the ‘New’ button:



This opens up a notebook, and you’re ready to program:



Note that a lot of functionality is available through to toolbar (i.e. inserting new cells (in ‘Insert’), deleting cells (in ‘Edit’), changing celltype (in ‘Cell’) and so on). A lot of keyboard shortcuts also exist (listed next to the function in the toolbar), but there also exist lists online, e.g. [this one](#), where especially the table ‘Command Mode - Esc to enter’ and ‘Whilst holding Entr - Running Cells’ are of special interest.

The perhaps most important part is that you execute a cell using **ctrl + enter** (execute cell and stay at current cell) or **shift + enter** (execute cell and proceed to next cell).

5 Environments with conda

Anaconda ships with a lot of things, including [conda](#), which is a package and environment manager. An environment contains information about what version of Python plus what packages and versions of these your project rely on. I believe that managing your Python and package versions is an important part of working on multiple projects simultaneously (or to be able to replicate old projects), which is why I’m introducing you to these.

It seems as if your computers are locked down, such that you are currently not able to utilize environments. As such this part is **OPTIONAL** and for informative purposes only.

This guide will cover two things:

1. How to create environments, and
2. How to activate environments.

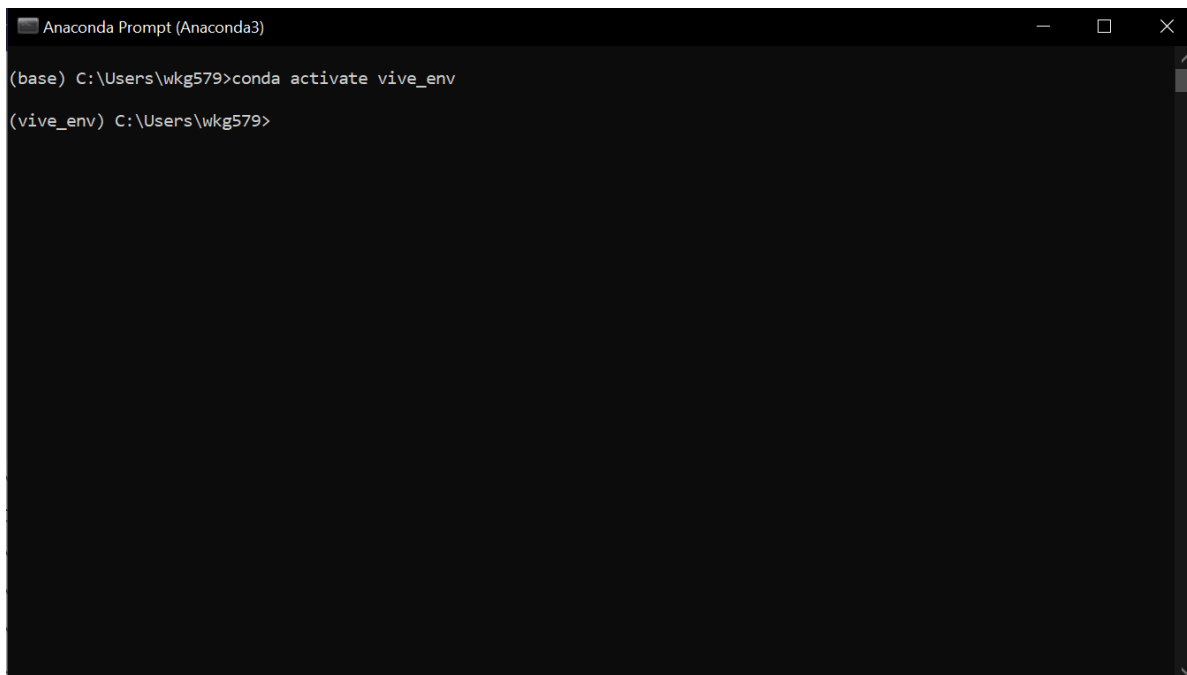
In addition, conda has a [guide on managing environments](#).

5.1 Creating environments

An environment can be created using the command `conda create -n my_env`, where `my_env` is the desired name of the environment. If you want to create an environment with a specific Python version (e.g. 3.10), you can utilize the command `conda create -n my_env python=3.10`.

5.2 Activating an environment

An environment can be activated using the command `conda activate my_env`, where `my_env` now is the name of a preexisting environment. The default environment is called `base`, which is why this is written to the left in the Anaconda Prompt. For this course, I have created an environment called `vive_env`, which I can activate:

A screenshot of the Anaconda Prompt terminal window. The title bar reads "Anaconda Prompt (Anaconda3)". The terminal shows the command `conda activate vive_env` being entered and executed. The prompt changes from `(base)` to `(vive_env)`, indicating the environment has been successfully activated. The current directory is `C:\Users\wkg579>`.

```
Anaconda Prompt (Anaconda3)
(base) C:\Users\wkg579>conda activate vive_env
(vive_env) C:\Users\wkg579>
```

Note how the environment changes from `base` to `vive_env`, which indicates that the environment has changed. You would now be able to navigate to your desired folder and open a notebook as described in the sections Anaconda Prompt and Launching Jupyter Notebook

6 Workflow

Whenever you wish to program in Python, a general workflow could be as follows:

1. Open your project folder
2. Copy the path
3. Open the Anaconda Prompt 3.1 (OPTIONAL) Change drive by writing the specific drive follow by colon (i.e. R: for the R-drive) 3.2 (OPTIONAL) Activate a project-specific environment using `conda activate my_env`
4. Change directory to the project folder using `cd project_path`, where you paste in the project path from step 2
5. Launch Jupyter Notebook using the command `jupyter notebook`