

# Software Installation Guide

Francisco Rowe, Dani Arribas-Bel

2020-10-02



# Contents



# Purpose

This guide provides step-by-step description on how to install and access Python from your own computer.

Select your Operating System and follow the steps.

# What is your operating system?

MacOS



Linux



Choose A1

Choose A2

**A1** MacOS Installation

**A2** Linux Installation

**A3** Windows 10 Pro/Student Installation

**A4** Windows Home or pre-10 Installation

*Need to find out your Windows version? Click [HERE](#)*



# MacOS Installation

## Installing Python

### Requirements

1. A stable internet connection
2. ~10GB of space on your hard drive
3. MacOS version 10.13 or newer i.e. High Sierra, Mojave or Catalina. If you are unsure what version you are running click on the apple icon in the top left of the screen and then **About this Mac**.
4. Mac hardware must be a 2010 model or newer

### Installation steps

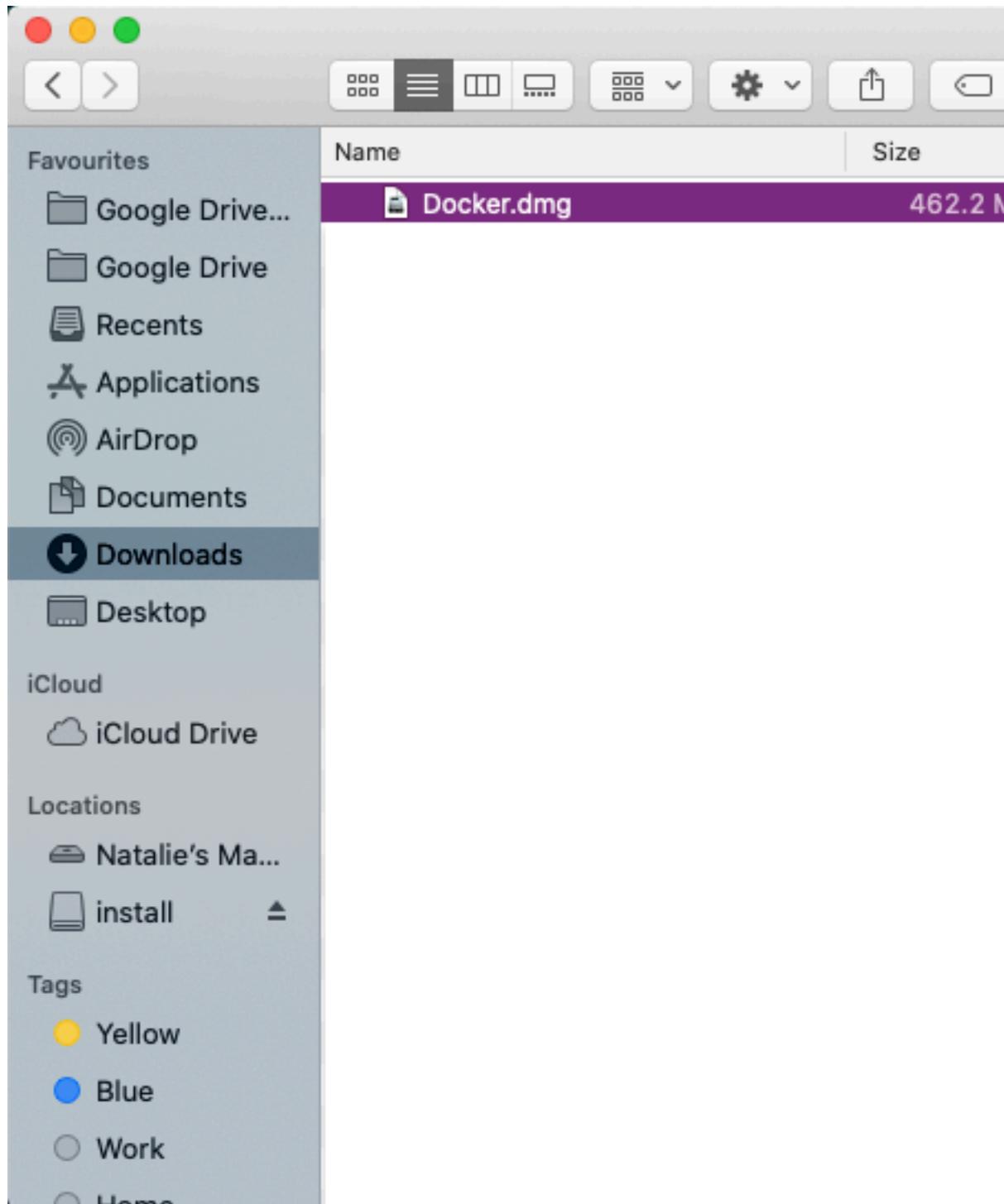
1. Go to the dockerhub website.
2. Ensure you meet the criteria for download (it is the same as stated above) and then select ‘Get Stable’ button.



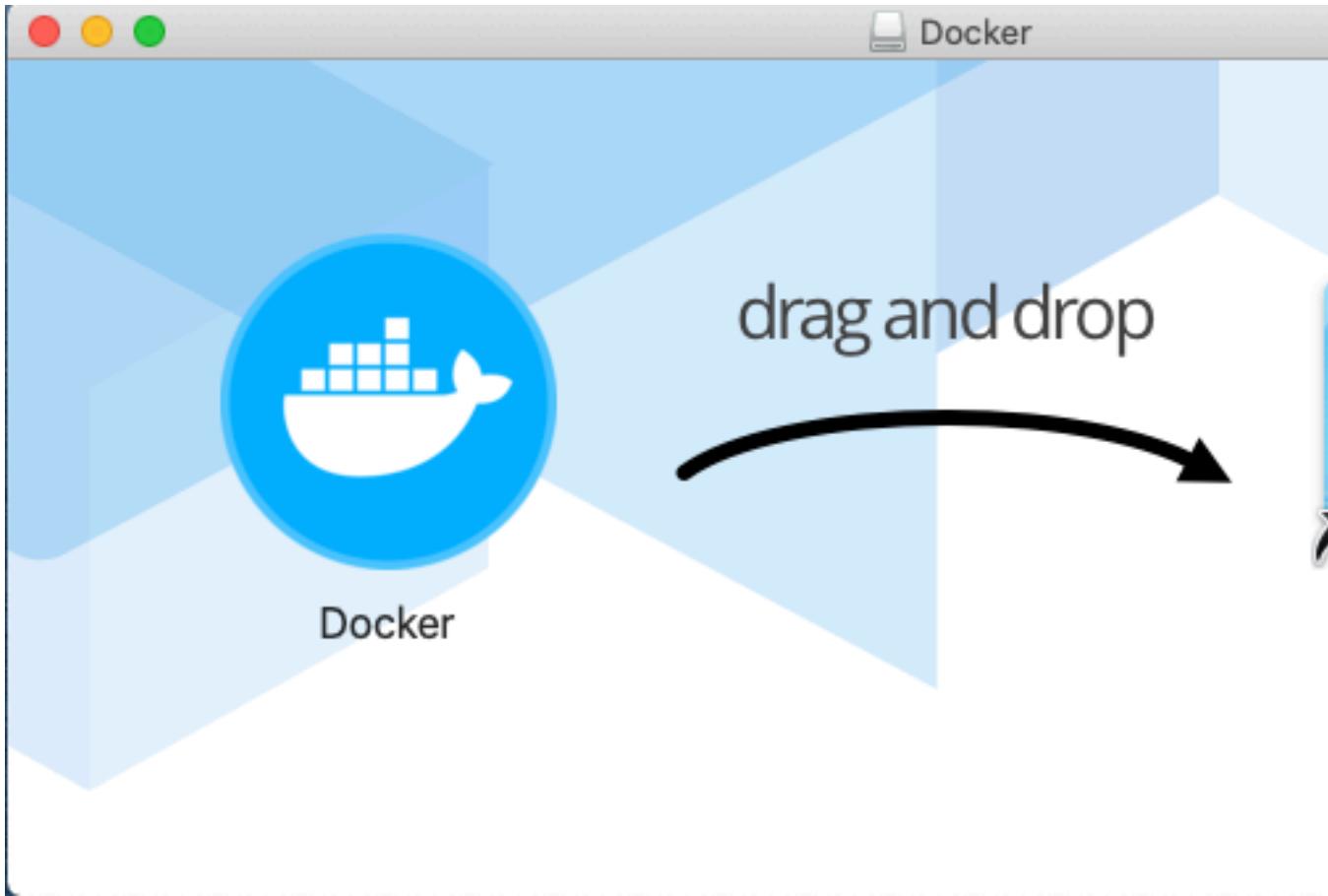
The screenshot shows the Docker Hub website interface. At the top, there's a blue header bar with the Docker Hub logo on the left and a search bar on the right containing the placeholder text "Search for great content (e.g., mysql)". Below the header, there are two main navigation links: "Explore" and "Docker Desktop for Mac". The "Docker Desktop for Mac" link is highlighted in blue, indicating it's the current page. The main content area features a large image of the Docker logo (a white whale carrying a stack of shipping containers) on a dark background. To the right of the image, the text "Docker Desktop for Mac" is displayed in a large, bold, dark font. Below that, the word "By" is followed by the "Docker" logo in blue. A descriptive subtitle reads "The fastest and easiest way to get started with". At the bottom of the main section, there are three small, light-colored buttons labeled "Edition", "macOS", and "x86-64".

3. This will then download to your machine but may take some time. Once finished, to access this download go to **Finder > Downloads > Docker.dmg** and double click.





4. You should then be prompted to drag and drop this application into the applications folder like so:



You may get further windows asking for access to the program. To these you can click **Open > Ok** > enter your account password and click **Install helper**

5. After you have done this, the whale icon should now show in your taskbar:



You have successfully downloaded Docker!

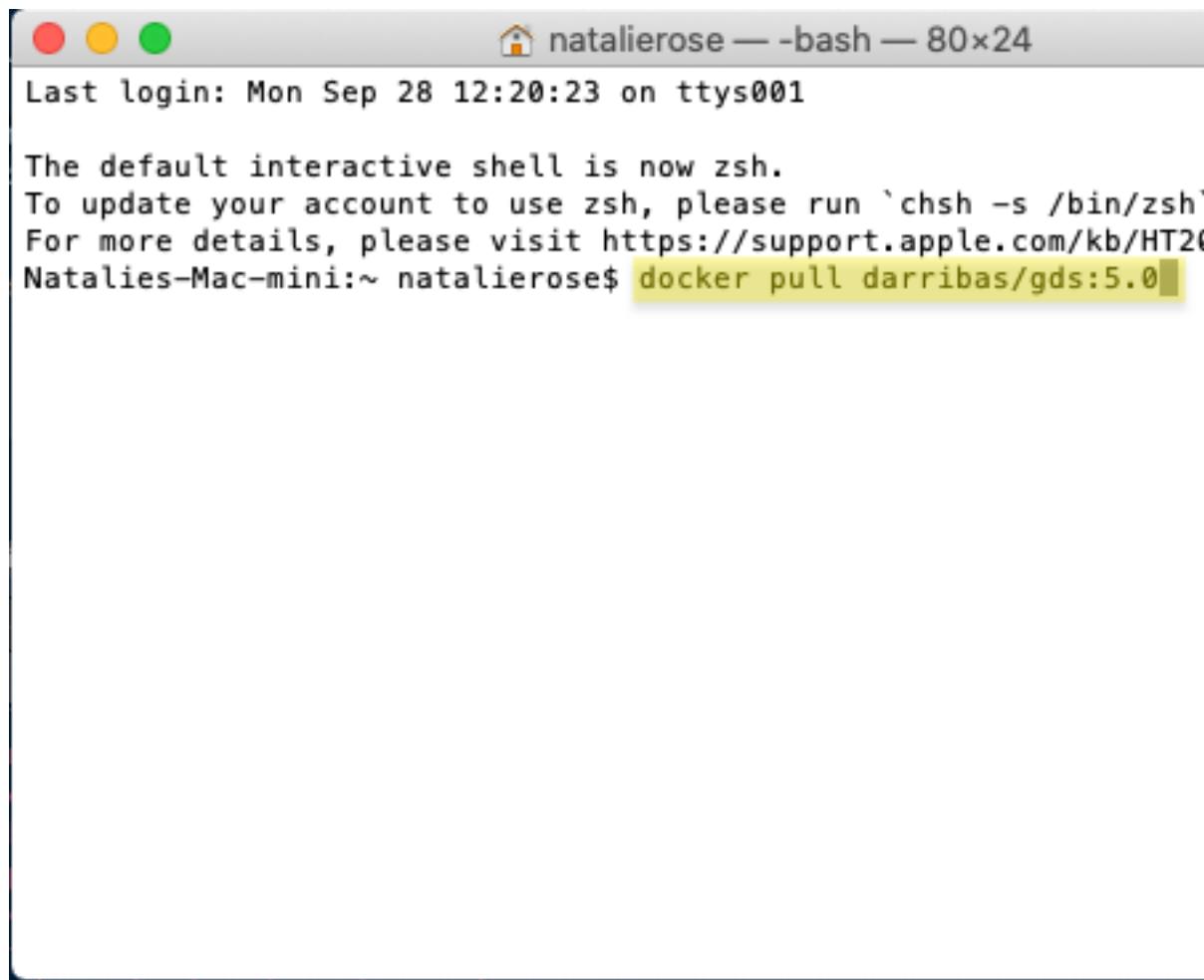
Next steps: Using Docker

## Using Docker

Now we have Docker installed we can use it to access Python and all the associated packages we need for the practicals

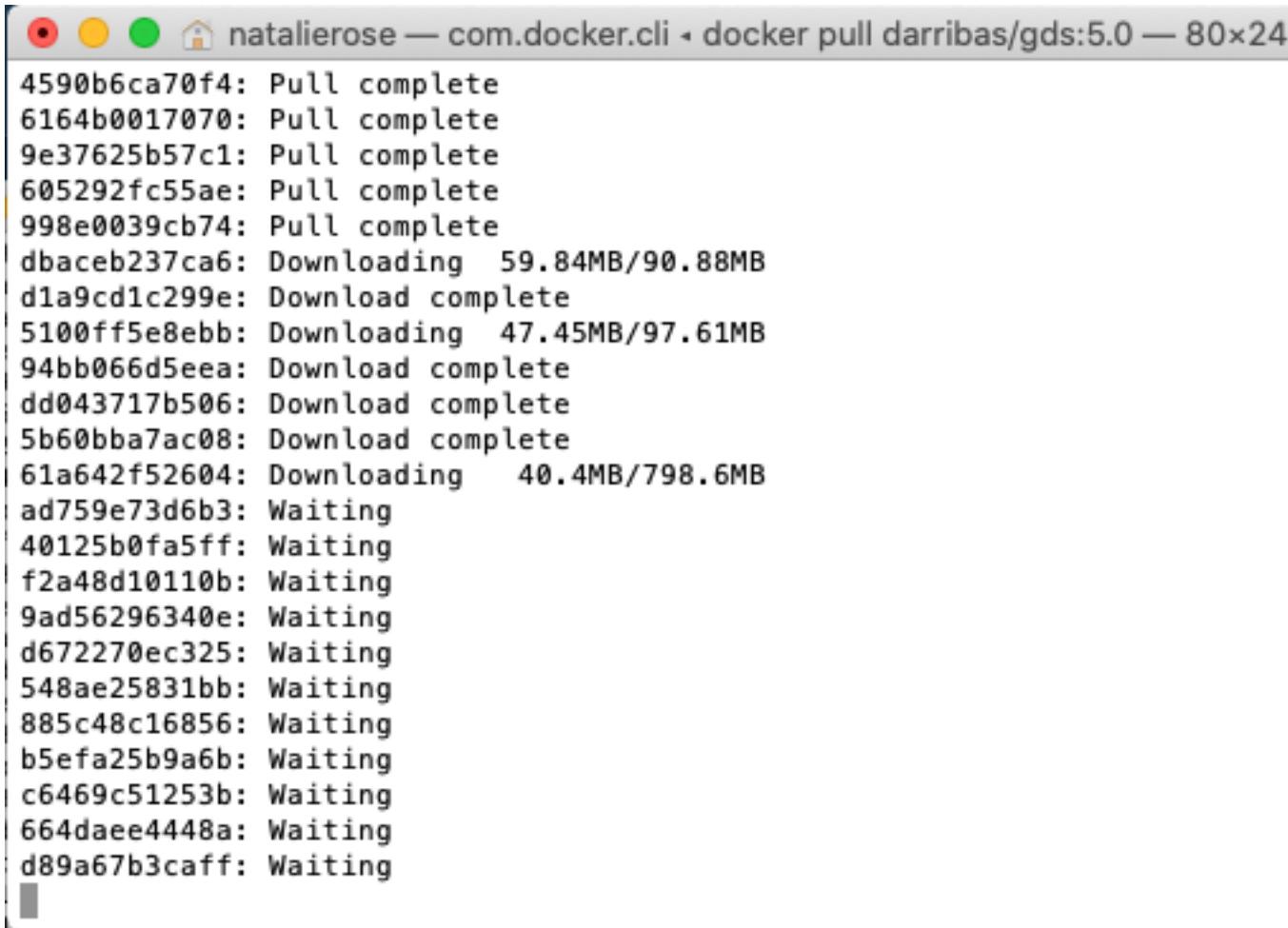
### Installing the GDS environment

1. Access your terminal: **Launchpad > Other > Terminal**
2. In a fresh line in the terminal type the following to install the GDS environment container: `docker pull darribas/gds:5.0`



```
natalierose — -bash — 80x24
Last login: Mon Sep 28 12:20:23 on ttys001
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`
For more details, please visit https://support.apple.com/kb/HT20
Natalies-Mac-mini:~ natalierose$ docker pull darribas/gds:5.0
```

3. This should now prompt a long download process that looks a bit like this:



The screenshot shows a terminal window titled 'natalierose — com.docker.cli' with the command 'docker pull darribas/gds:5.0'. The window displays a list of Docker image IDs and their current status:

- 4590b6ca70f4: Pull complete
- 6164b0017070: Pull complete
- 9e37625b57c1: Pull complete
- 605292fc55ae: Pull complete
- 998e0039cb74: Pull complete
- dbaceb237ca6: Downloading 59.84MB/90.88MB
- d1a9cd1c299e: Download complete
- 5100ff5e8ebb: Downloading 47.45MB/97.61MB
- 94bb066d5eea: Download complete
- dd043717b506: Download complete
- 5b60bba7ac08: Download complete
- 61a642f52604: Downloading 40.4MB/798.6MB
- ad759e73d6b3: Waiting
- 40125b0fa5ff: Waiting
- f2a48d10110b: Waiting
- 9ad56296340e: Waiting
- d672270ec325: Waiting
- 548ae25831bb: Waiting
- 885c48c16856: Waiting
- b5efa25b9a6b: Waiting
- c6469c51253b: Waiting
- 664daee4448a: Waiting
- d89a67b3caff: Waiting

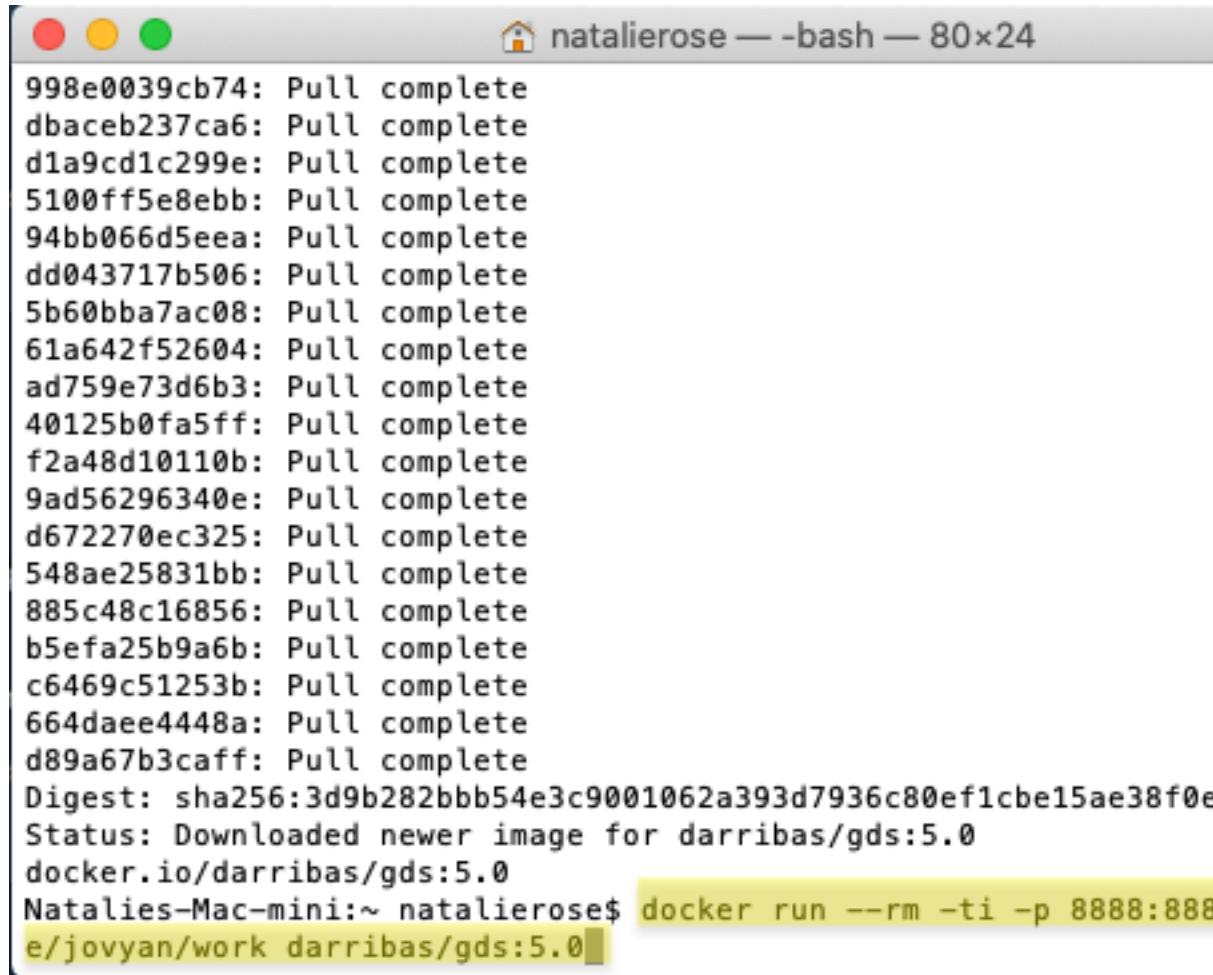
Dont be alarmed if it seems to take a very long time.

You will know this has completed when each line says 'Pull complete' and the new line gives your machine name followed by a \$ sign.

## Running Python

### Running the container

1. In the new terminal line type the following command to run the container:  
docker run --rm -ti -p 8888:8888 -v \${PWD}:/home/jovyan/work  
darribas/gds:5.0



```
998e0039cb74: Pull complete
dbaceb237ca6: Pull complete
d1a9cd1c299e: Pull complete
5100ff5e8ebb: Pull complete
94bb066d5eea: Pull complete
dd043717b506: Pull complete
5b60bba7ac08: Pull complete
61a642f52604: Pull complete
ad759e73d6b3: Pull complete
40125b0fa5ff: Pull complete
f2a48d10110b: Pull complete
9ad56296340e: Pull complete
d672270ec325: Pull complete
548ae25831bb: Pull complete
885c48c16856: Pull complete
b5efa25b9a6b: Pull complete
c6469c51253b: Pull complete
664daeee4448a: Pull complete
d89a67b3caff: Pull complete
Digest: sha256:3d9b282bbb54e3c9001062a393d7936c80ef1cbe15ae38f0e
Status: Downloaded newer image for darribas/gds:5.0
docker.io/darribas/gds:5.0
Natalies-Mac-mini:~ natalierose$ docker run --rm -ti -p 8888:8888
e/jovyan/work darribas/gds:5.0
```

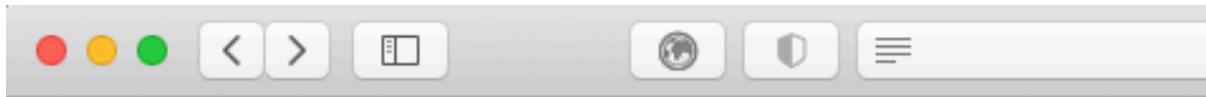
You have now started a Python session.

**NOTE:** It is important that you do not close the terminal window until you are finished in this Python session

2. To access this session go to your chosen web browser (e.g. Safari/Chrome) and type: `localhost:8888` into the search bar
3. The page that loads will prompt you for a password. This password can be found in the text in the terminal following the last command you ran (step 9). A long series of numbers and letters will be preceded by `?token=`. Copy this long series of characters and paste into the password box in your browser.

```
natalierose — com.docker.cli - docker run --rm -ti -p 8888:8888 -v /Us
kernels (twice to skip confirmation).
[C 10:12:45.563 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/jovyan/.local/share/jupyter/runtime/nbserver-6-open
    Or copy and paste one of these URLs:
        http://cee202bbc035:8888/?token=a0c182d2e15922a6e7ec4a237780c9de
167d3d6e
        or http://127.0.0.1:8888/?token=a0c182d2e15922a6e7ec4a237780c9dddf
d3d6e
[W 10:12:59.367 NotebookApp] Clearing invalid/expired login cookie user@lhost-8888
[W 10:12:59.372 NotebookApp] Clearing invalid/expired login cookie user@lhost-8888
[I 10:13:52.083 NotebookApp] Build is up to date
[I 10:13:56.551 NotebookApp] Creating new notebook in /
[I 10:13:56.558 NotebookApp] Saving Untitled.ipynb
[I 10:13:56.572 NotebookApp] Writing notebook-signing key to /home/jovyan/.share/jupyter/notebook_secret
-[I 10:13:57.163 NotebookApp] Kernel started: c47336c5-8b44-446f-b062-a6e7
[I 10:16:01.490 NotebookApp] Saving file at /Untitled.ipynb
[I 10:16:01.493 NotebookApp] Saving Untitled.ipynb
```



[Password or token](#)

## Token authentication

If no password has been specified, Jupyter Notebook will look for its login token in the URL. You can either [enable a password](#), or

The command:

```
jupyter notebook
```

will show you the URLs of all notebooks. Copy and paste into your browser.

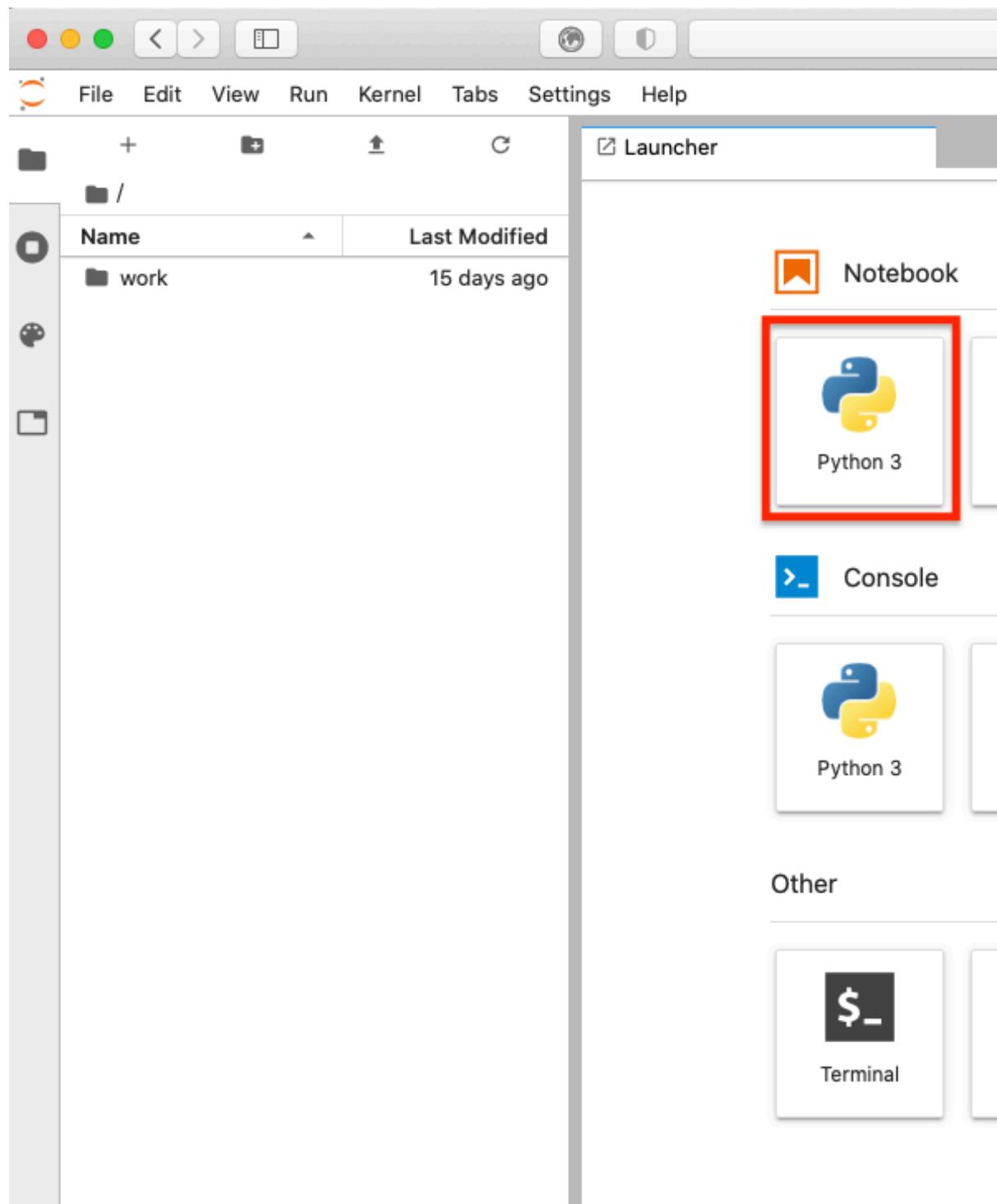
```
Currently running at:  
http://localhost:8888  
in directory /Users/username/notebooks
```

or you can paste just the URL:

See [the documentation](#) for more information about token authentication, if you would like to use it.

Cookies are required for authentication.

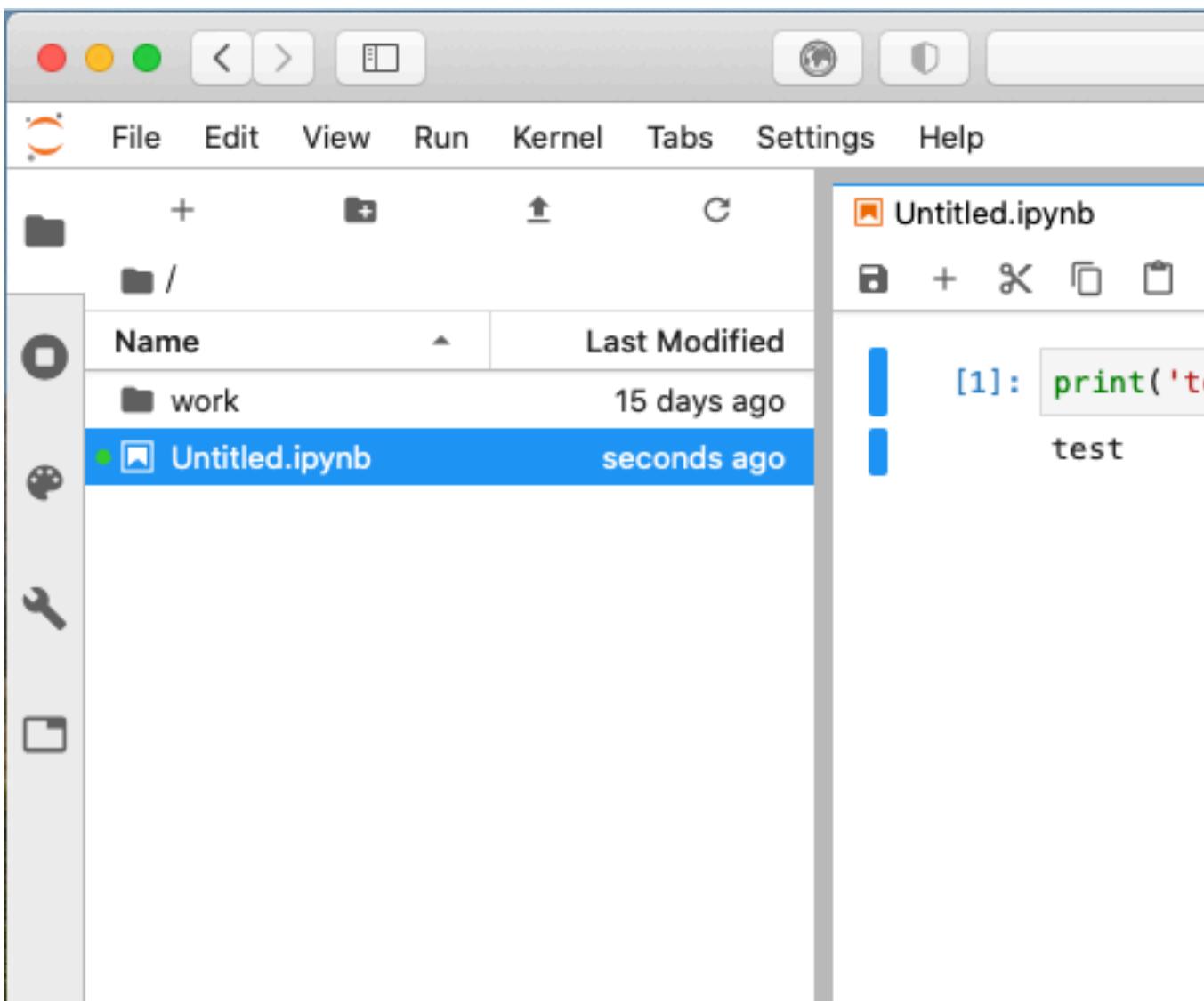
4. Now you are in Jupyter Lab you can open up a Python 3 notebook



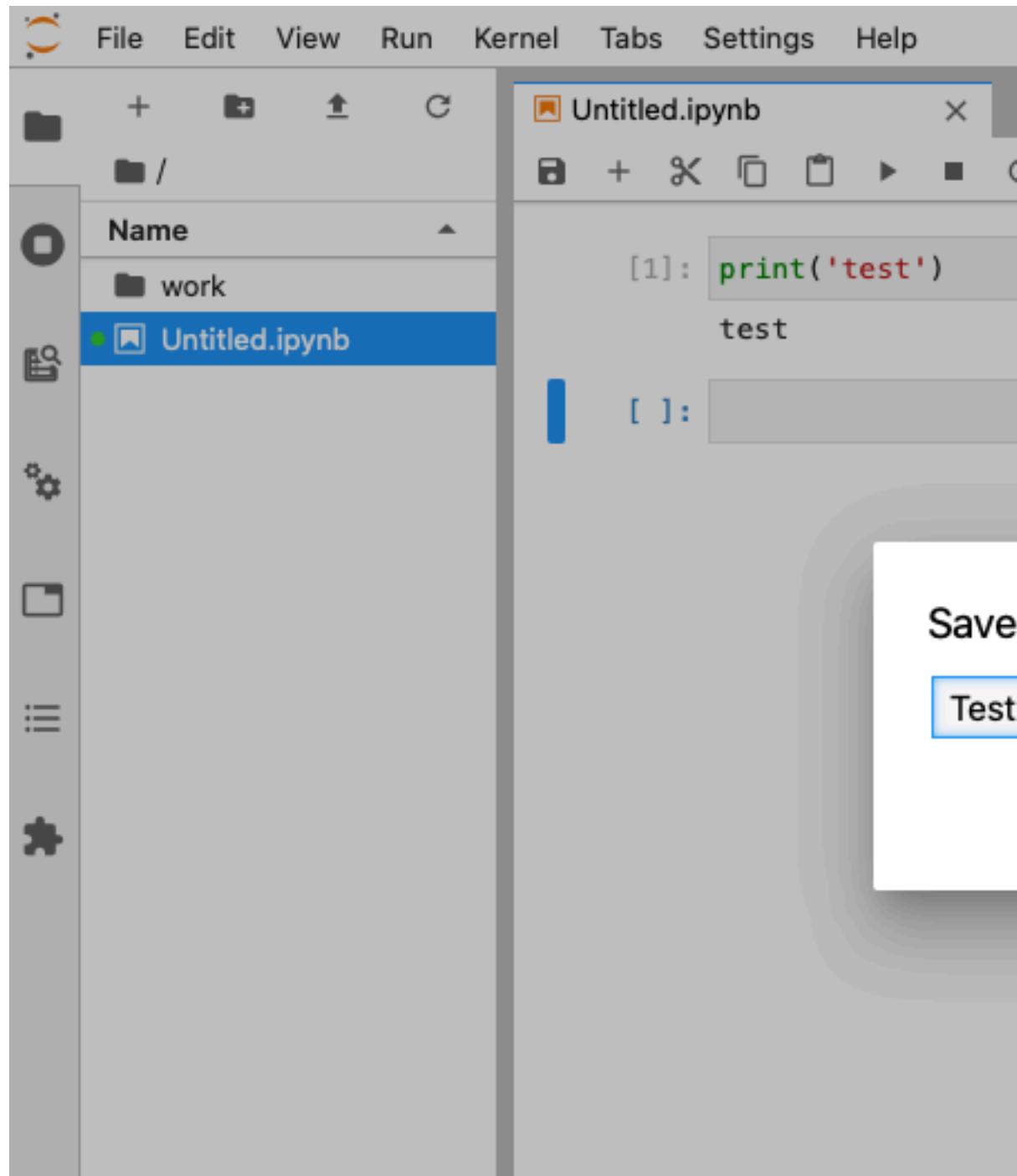
## Using Jupyter Notebook

### Using Jupyter Notebook

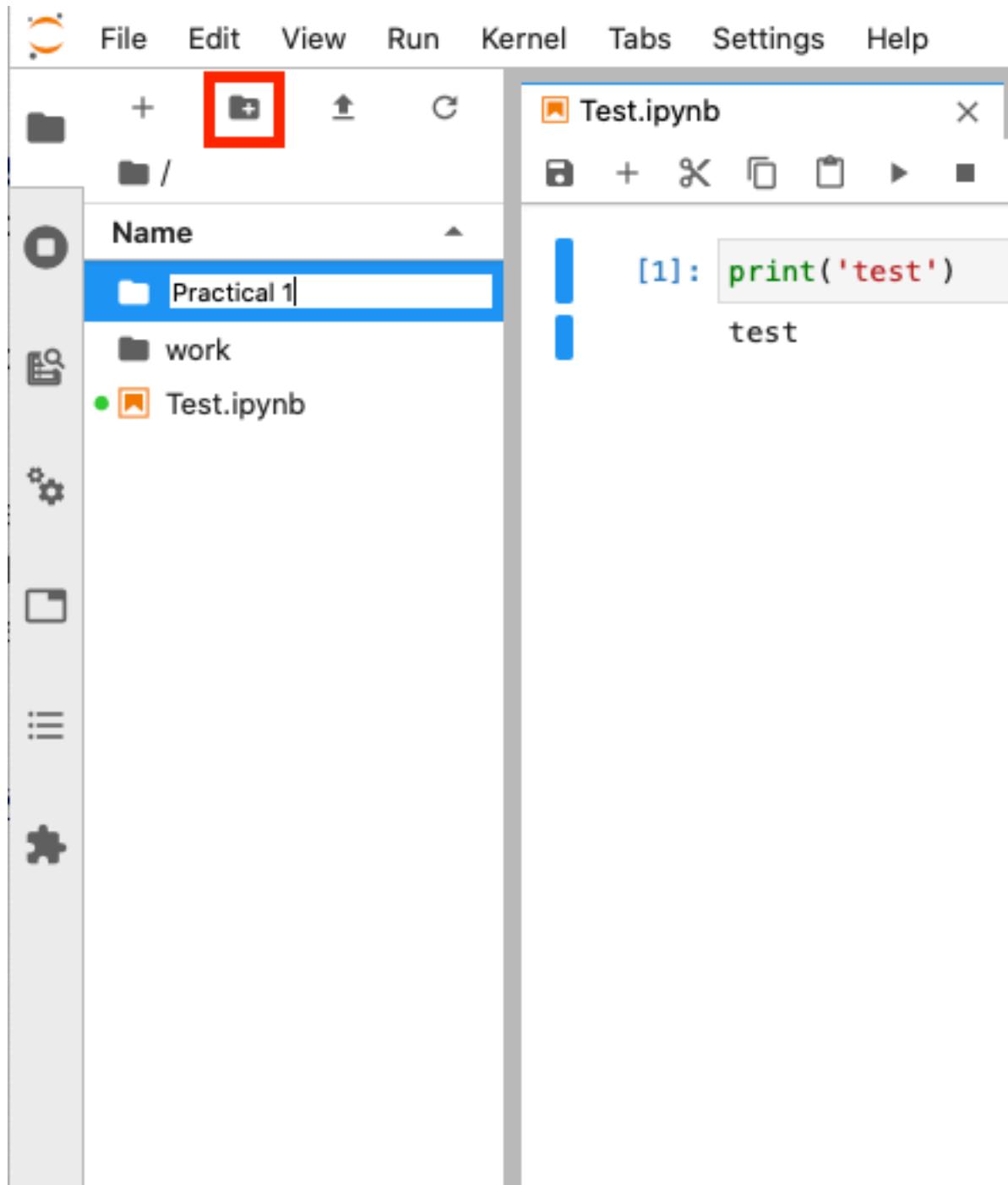
- This notebook is where you will run your code. Each shaded box is called a kernel. To test this out you can type `print('test')` into one of these kernels. To run the code use the shortcut **Ctrl + Enter**.



- You can save your notebook using **File > Save notebook as**

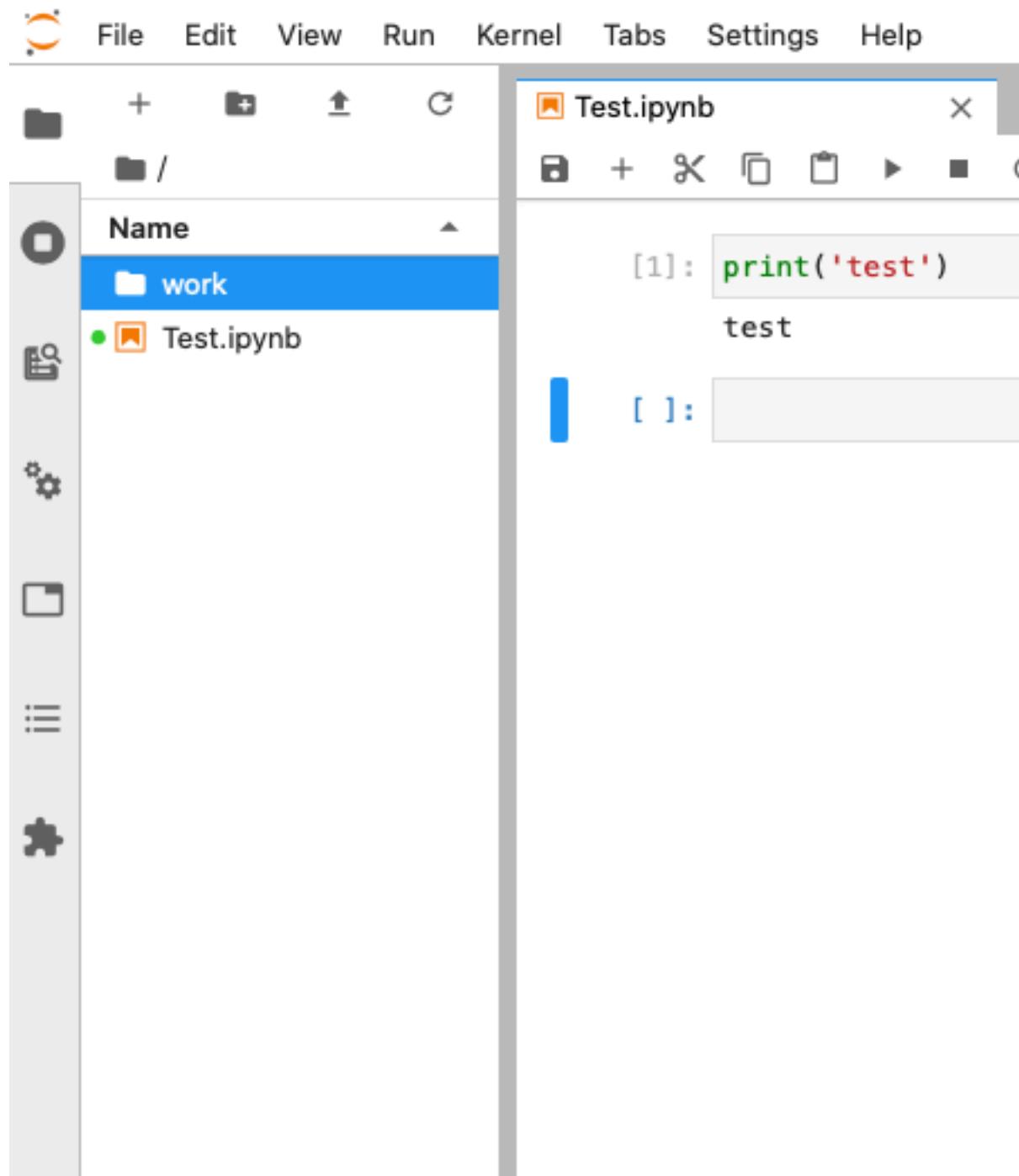


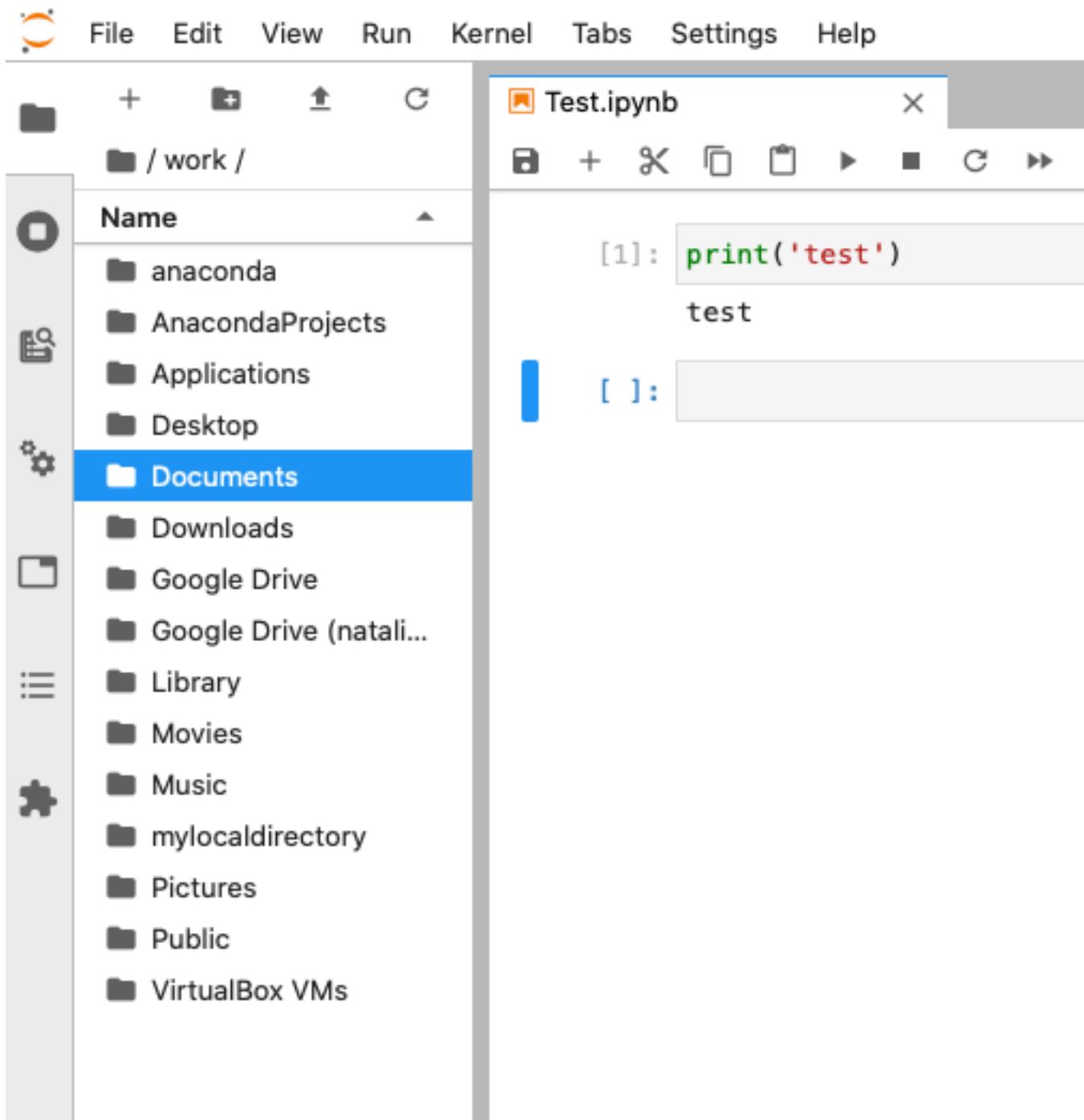
- You can create new folders to organise your work

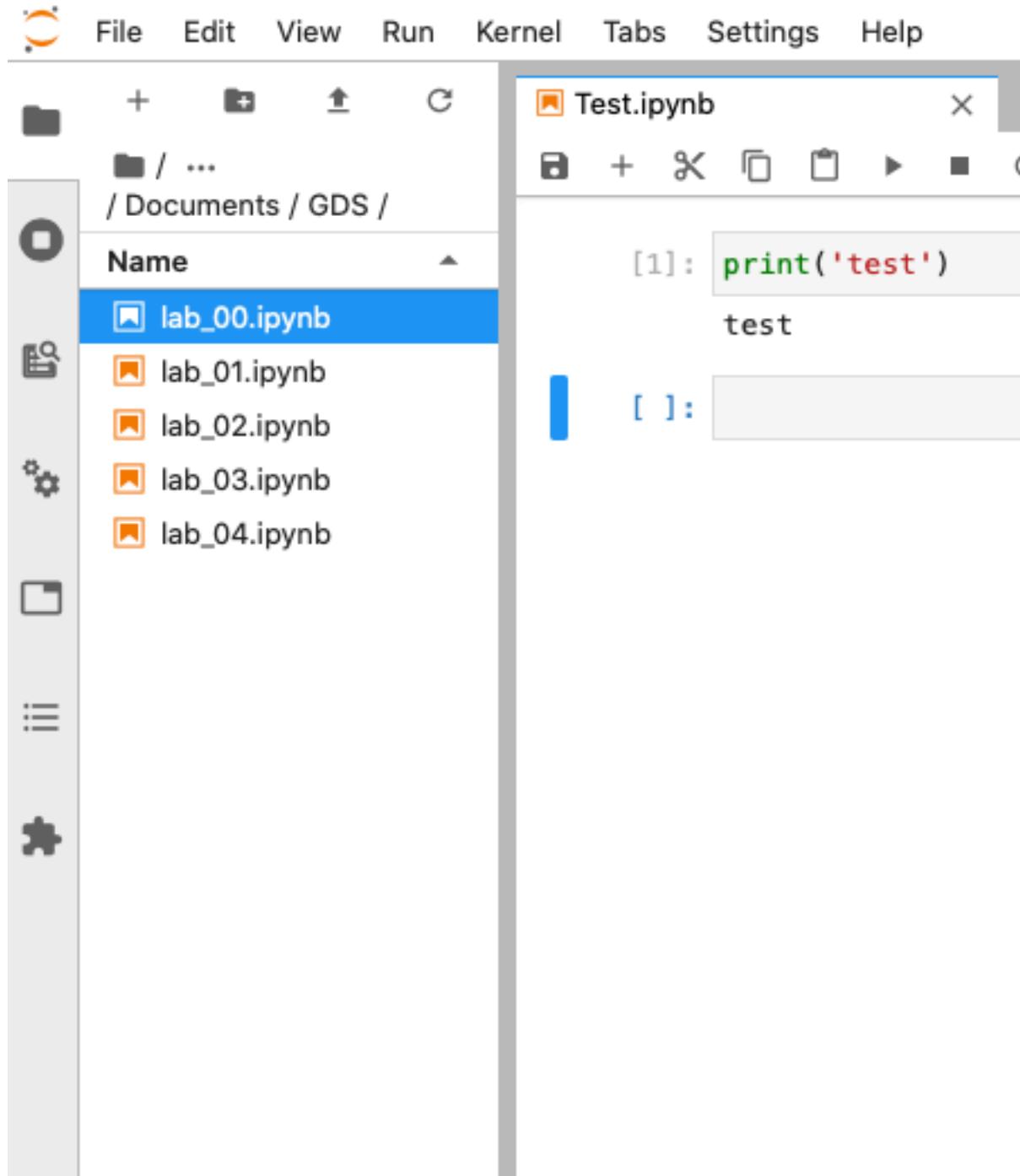


- And you can access other files on your machine through the ‘Work’ folder

in the File Browser. From here you can navigate to your Documents and designated folder for this module







Ending your session

## Ending your session

Once you have finished in your Jupyter session and have saved all your work, you can end the session from the terminal.

Using `Ctrl + C` will prompt a `y/n` option. Either type `y` or `Ctrl + C` again to end the session.

```
natalierose — com.docker.cli - docker run --rm -ti -p 8888:8888 -v /Us
[W 11:41:17.975 NotebookApp] No such file or directory: Test.ipynb
[W 11:41:17.976 NotebookApp] 404 GET /api/contents/Test.ipynb?1600947677
17.0.1) 2.50ms referer=http://localhost:8888/lab?
[W 11:41:18.073 NotebookApp] 404 GET /api/contents/Test.ipynb?content=08
8050 (172.17.0.1): No such file or directory: Test.ipynb
[W 11:41:18.074 NotebookApp] No such file or directory: Test.ipynb
[W 11:41:18.075 NotebookApp] 404 GET /api/contents/Test.ipynb?content=08
8050 (172.17.0.1) 2.90ms referer=http://localhost:8888/lab?
[I 11:41:18.158 NotebookApp] Uploading file to /Test.ipynb
[I 11:41:23.831 NotebookApp] Creating new directory in /
^C[I 11:46:37.146 NotebookApp] interrupted
Serving notebooks from local directory: /home/jovyan
1 active kernel
The Jupyter Notebook is running at:
http://198745c04037:8888/?token=bb9e07801410ccdd27577dab0aeee166d0ee38c17
or http://127.0.0.1:8888/?token=bb9e07801410ccdd27577dab0aeee166d0ee38c17
Shutdown this notebook server (y/[n])? No answer for 5s: resuming operat
[W 11:46:58.409 NotebookApp] delete /Practical 1
[I 11:47:00.080 NotebookApp] Creating new directory in /
^C[I 11:47:13.992 NotebookApp] interrupted
Serving notebooks from local directory: /home/jovyan
1 active kernel
The Jupyter Notebook is running at:
http://198745c04037:8888/?token=bb9e07801410ccdd27577dab0aeee166d0ee38c17
or http://127.0.0.1:8888/?token=bb9e07801410ccdd27577dab0aeee166d0ee38c17
Shutdown this notebook server (y/[n])?
```

You can now safely shut the terminal window.

Next time you go to run a Jupyter Notebook you will not need to repeat the whole process as you have already installed Docker and the GDS environment. Instead you can start from Running the container and carry on from there.



# **Linux Installation**

Instructions to install, test and run `gds_env` with Docker for .

For each of the sections, add:

Step-by-step instructions

Insert screenshots for each step

Insert a video with instructions

## **Installing Docker**

Draw on instructions from here and here and here

### **Requirements**

**Install steps**

**Check success**

## **Running Python through Docker**



# Windows 10 Pro/Student Installation



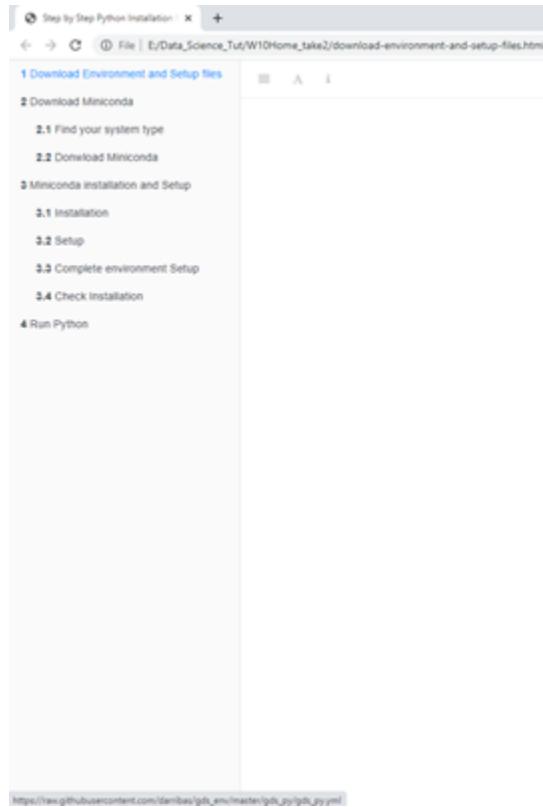
# **Windows Home or pre-10 Installation**

## **Installing Python**

### **Environment File**

We will start by downloading an environment file, which will later on install all packages that are relevant for your coding

**NOTE:** It would be best to create a dedicated folder (e.g. GDS\_2020) for this module.



**Step by Step Python Installation for Windows 10 Home**

Dominik Fahrner  
2020-09-28

## 1 Download Environment and Setup files

- To download the environment file, right click on this file

**gds\_py.yml**

- Select *Save link as...*

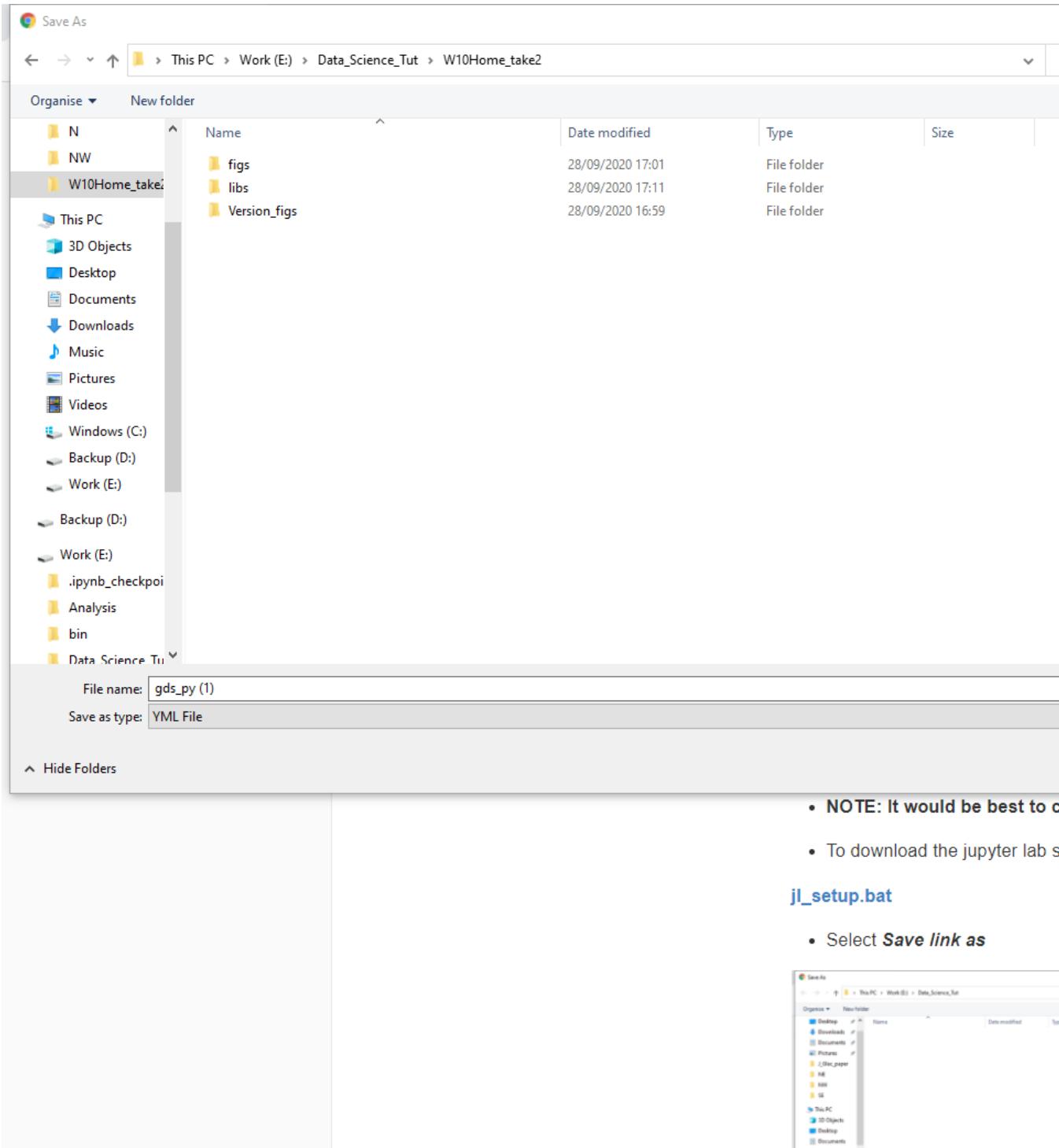
**JL\_setup.bat**

- Select *Save link as...*

- To download the environment file, right click on this file

### gds\_py.yml

- Select *Save link as...*.



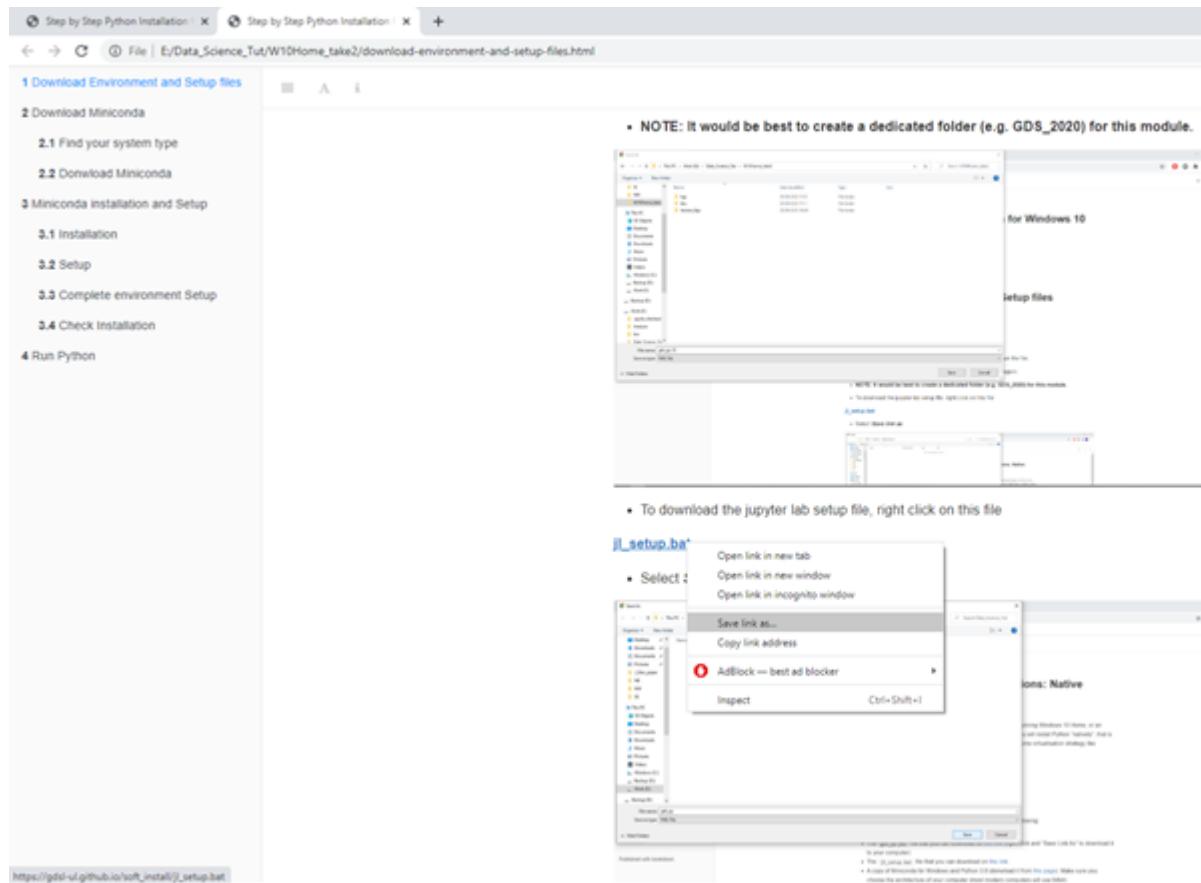
- A new window will pop up for saving the file, click on *save* to save the file.
- Make sure to save the file to a location that you can easily find again.

## User interface

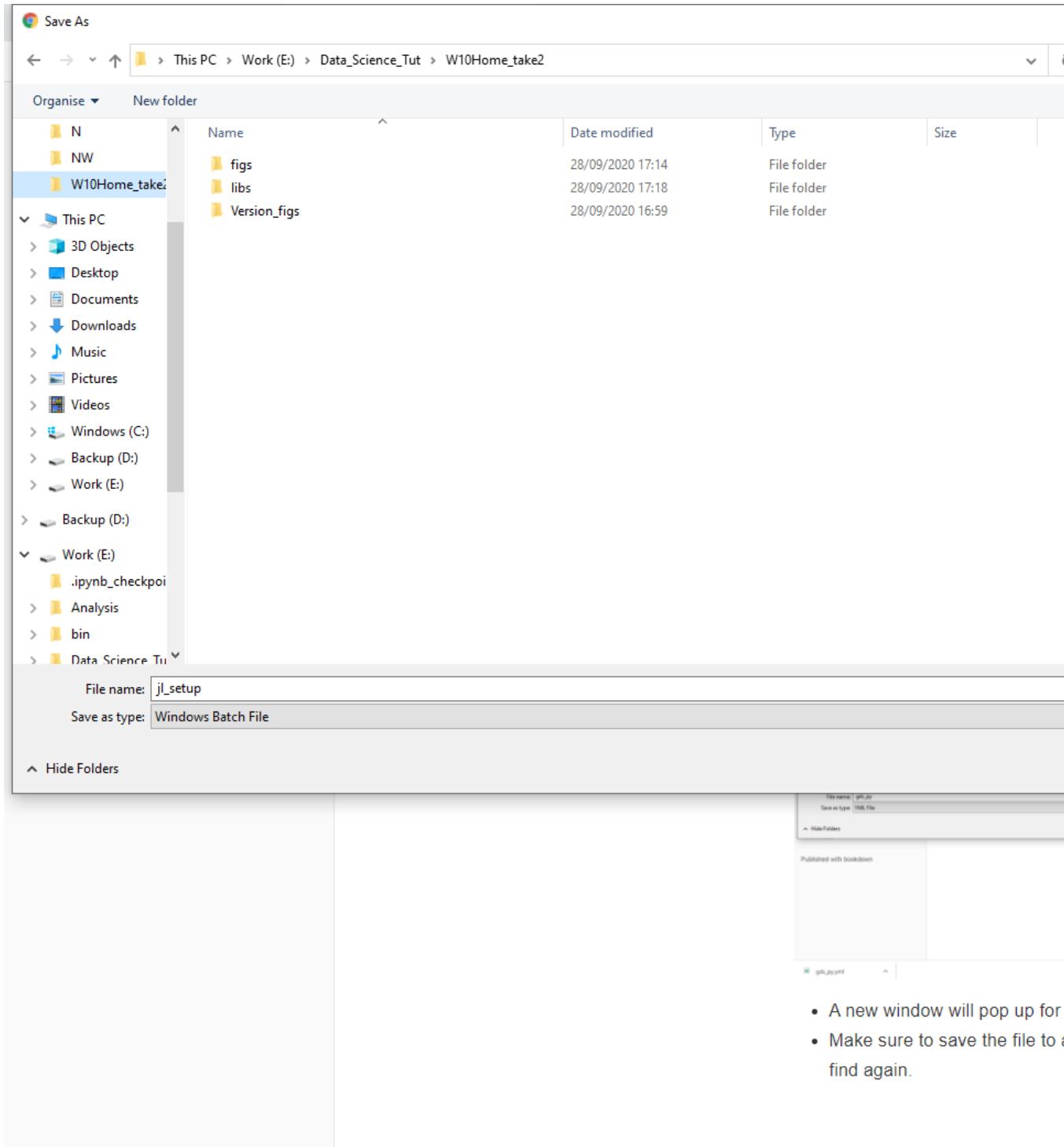
We will now download the file that will later setup your coding interface (which is called Jupyter Lab).

- To download the jupyter lab setup file, right click on this file

**jl\_setup.bat**



- Select *Save link as*

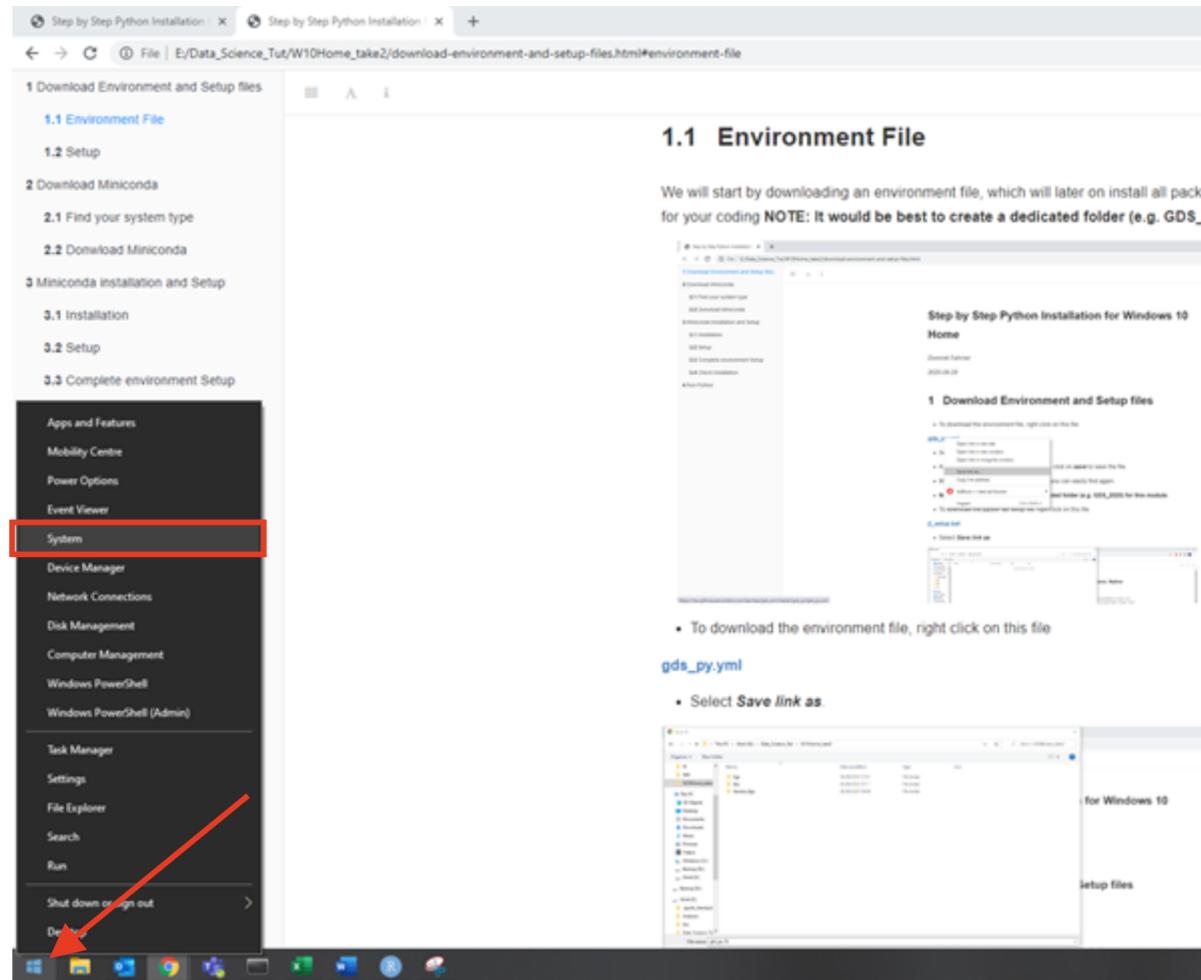


- A new window will pop up for saving the file, click on *save* to save the file.
- Make sure to save the file to a location (the same location as the `gds_py.yml` file) that you can easily find again.

## Download Miniconda

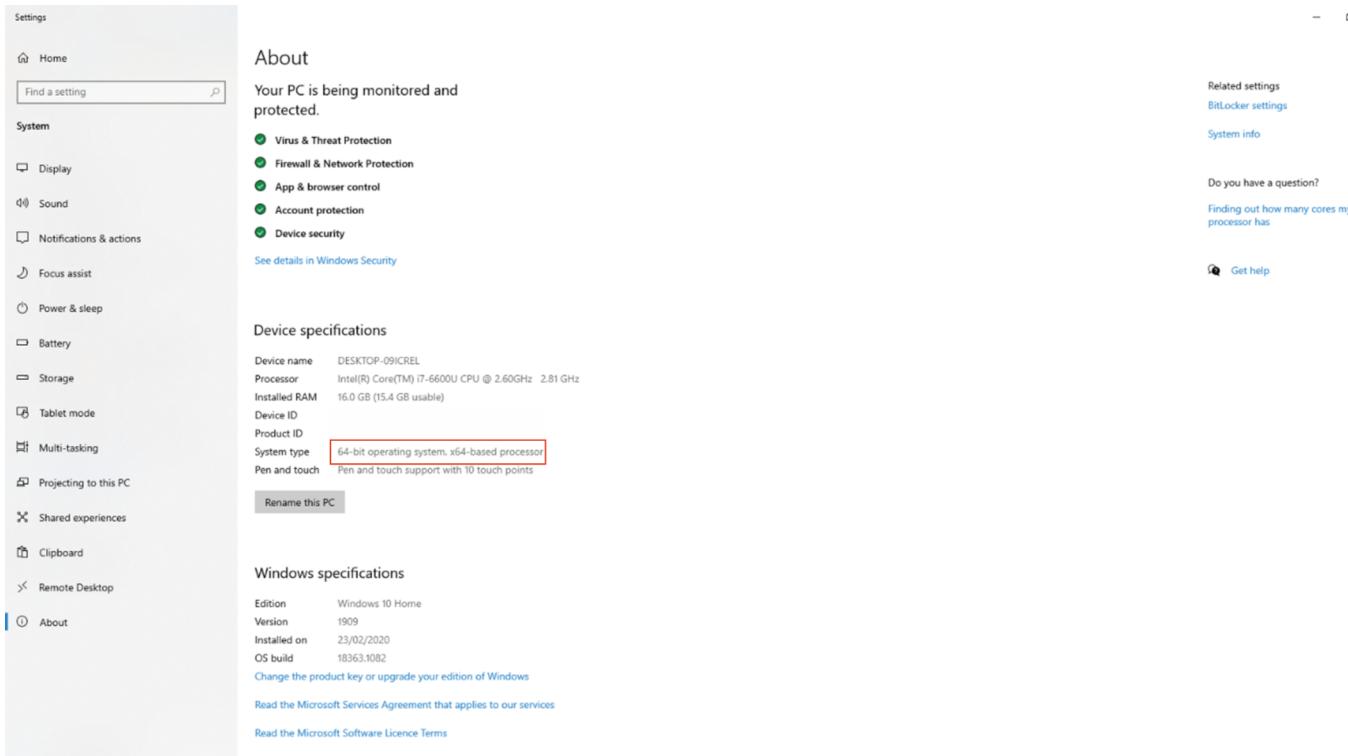
Find your system type

Before you can download Miniconda (which is a version of Anaconda), you need to find out what type your Windows system is. It can either be 32 bit or 64 bit (most modern computers use 64 bit).



- Right click on the windows logo in the left bottom corner of the task menu

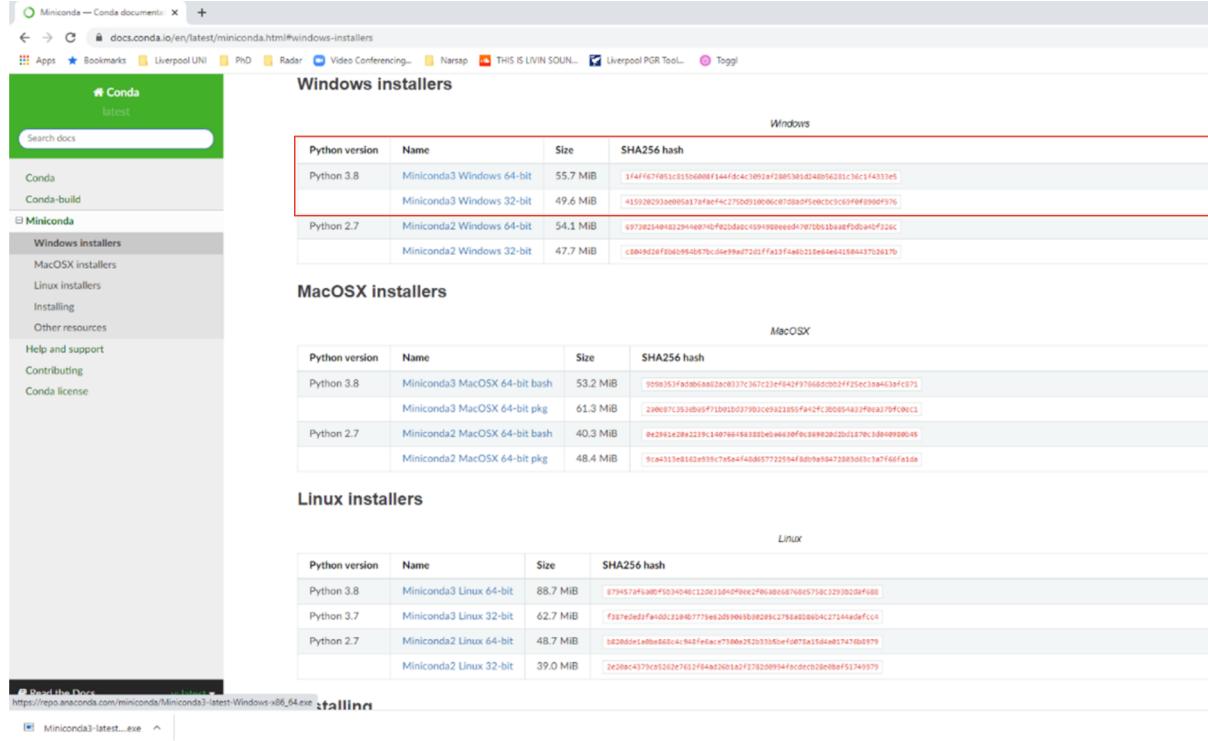
and select **System**



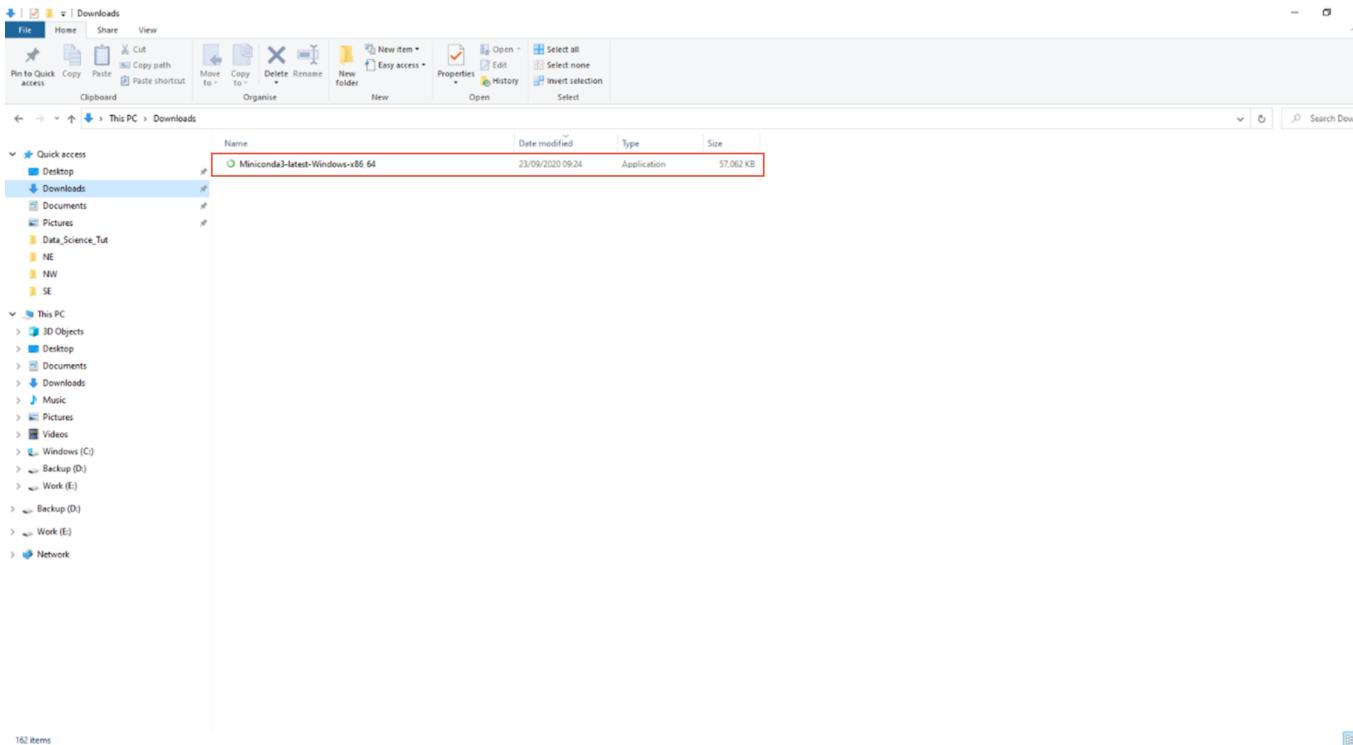
- This will bring you to your system information page.
- Look at the **System type** section and check if your operating system is 64-bit or 32-bit (highlighted in red).

## Download Miniconda

- Continue with opening this link to **Miniconda** by right clicking on the link and then selecting ***Open in new tab***.



- This will bring you to the Miniconda download page shown above.
- You have installation files for two different Python version (2.7 and 3.8) and for two different Windows systems (32 bit and 64 bit).
- We are using **Python 3.8**, so depending on which windows version you are using (32-bit or 64-bit), click on the relevant file in the Python 3.8 section (highlighted in red).
- This will download the Miniconda installation file to your **Downloads folder**.

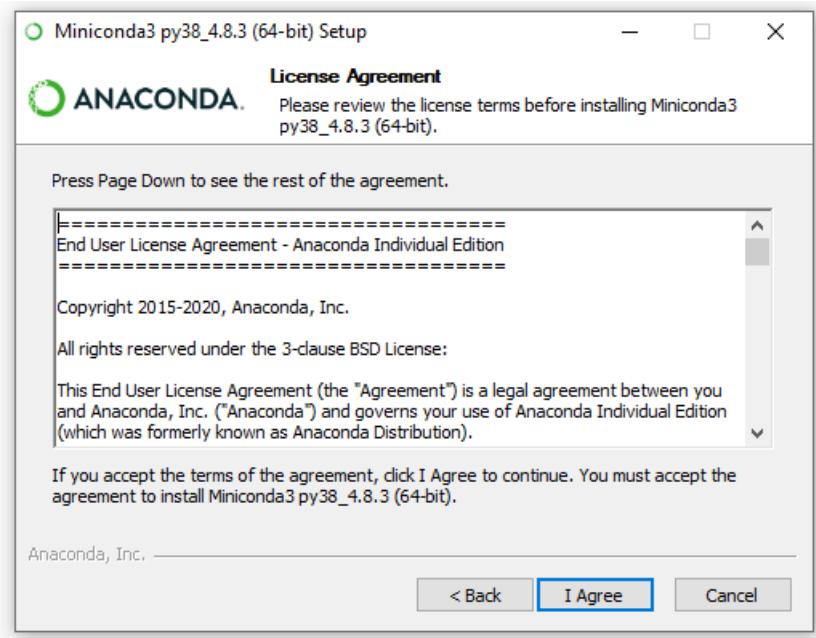


- Once the download has finished, navigate to your **Downloads folder** on your computer and double click on the *Miniconda3-latest-Windows-x86\_64* file to start the installation.
- Note: Double-check that you are installing the right version for your system.

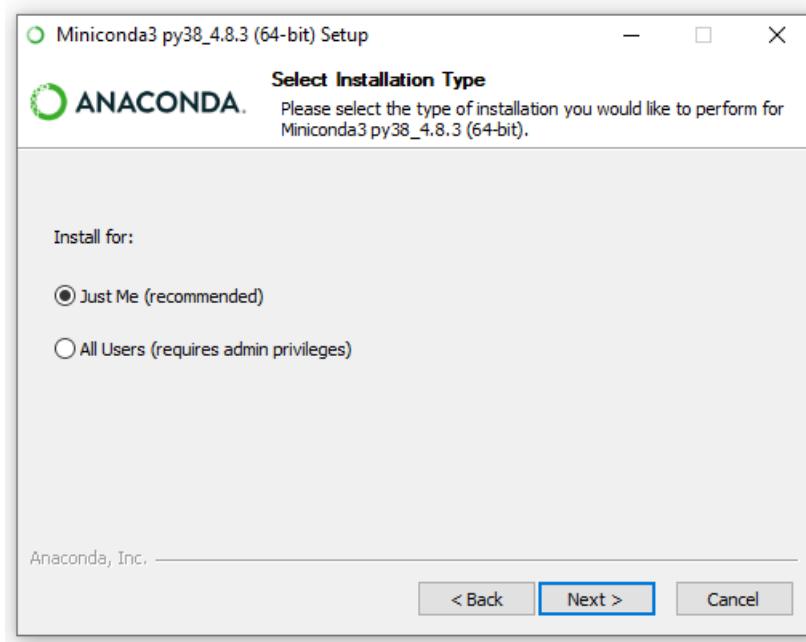
## Installing Minicoda



- Double clicking the downloaded file will open an installation window.
- Click ***Next*** on the first step.

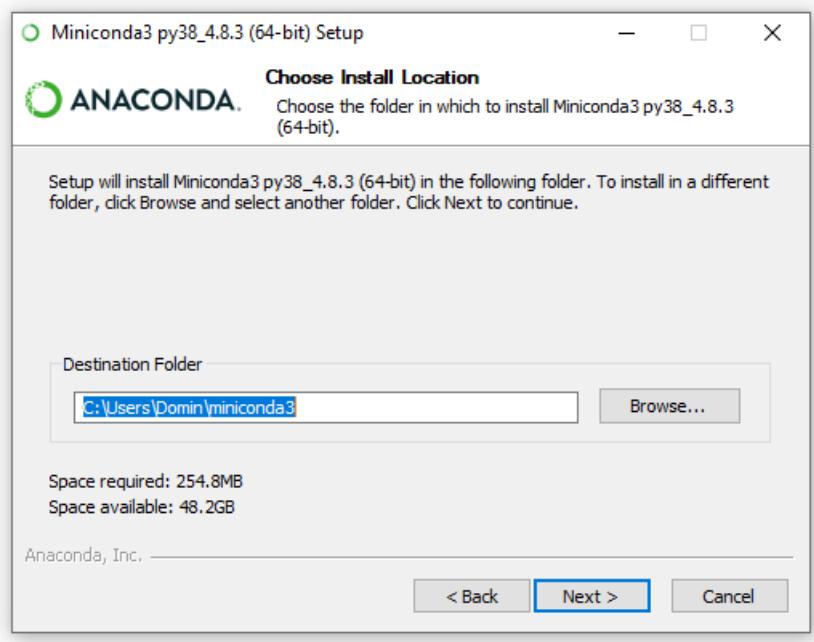


- Click **I Agree** in the next step which is the Terms and Conditions.

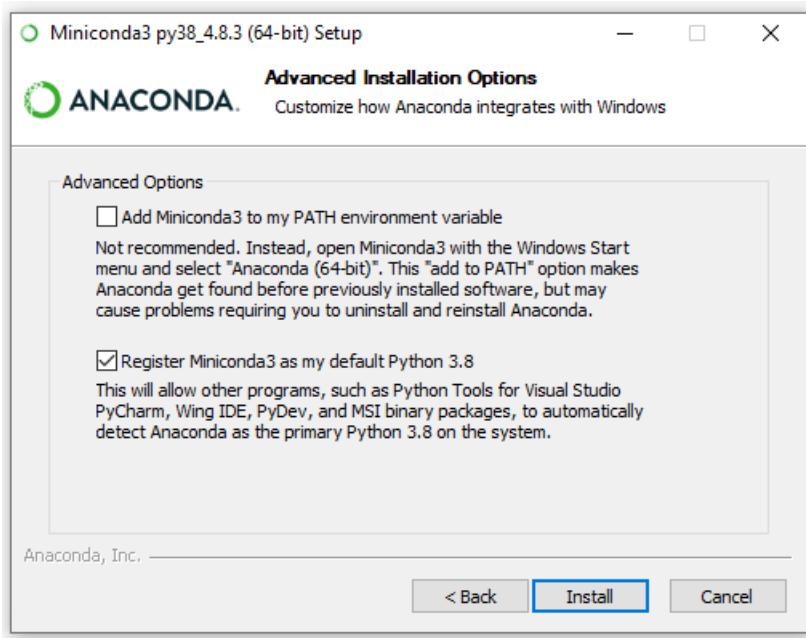


- In the next window, you can select if you want to install Miniconda for all users or just you.

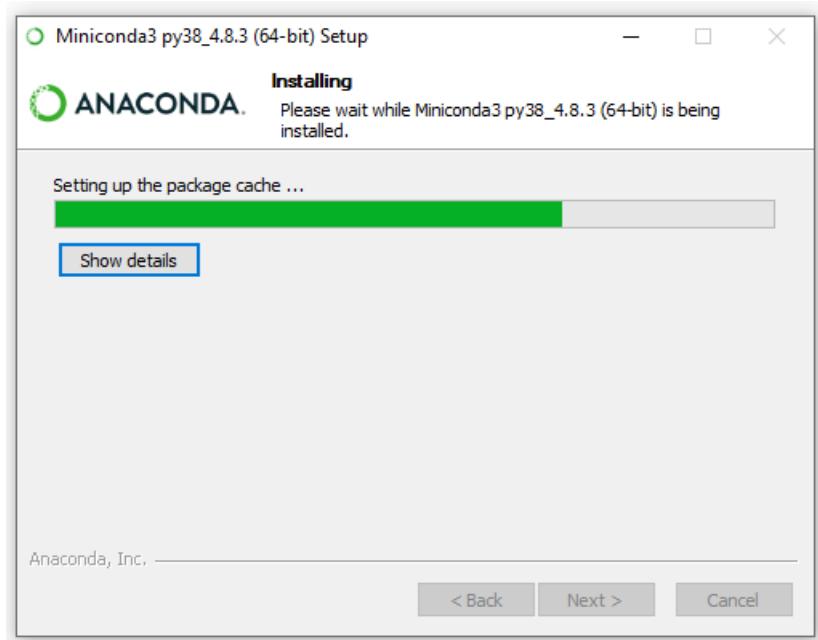
- Check that ***Just Me*** is selected and click next.



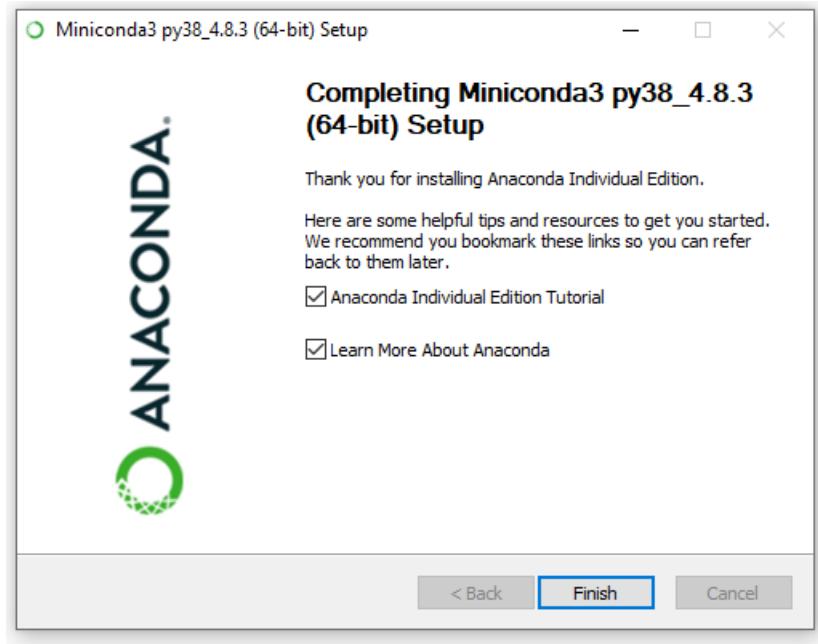
- The next window will ask you where to install Miniconda.
- Leave the path (highlighted in blue) as is and click next.



- The next window can be used for an advanced setup
- Leave the default settings as they are (Box ticked at *Register Miniconda3 as my default Python 3.8*).

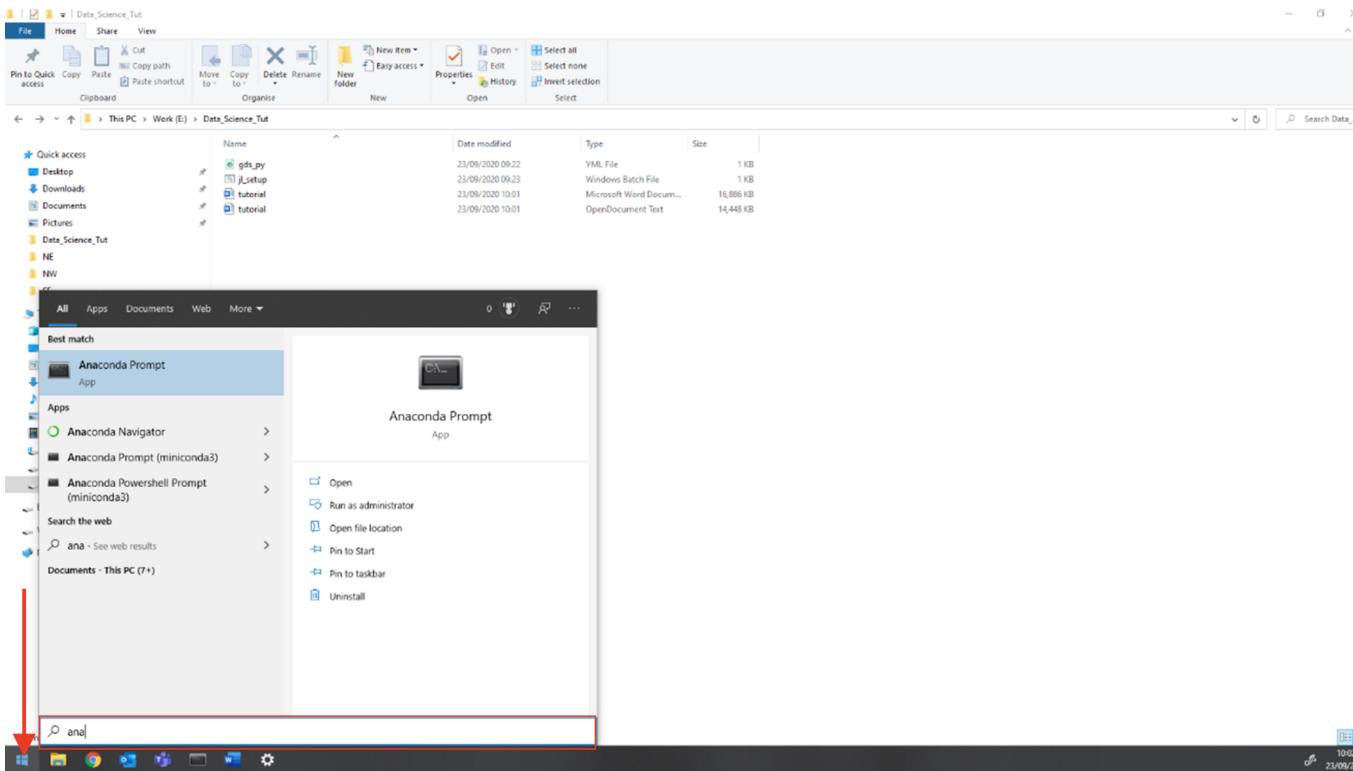


- Miniconda is now installing.
- Once the installation is complete, click **Next**.

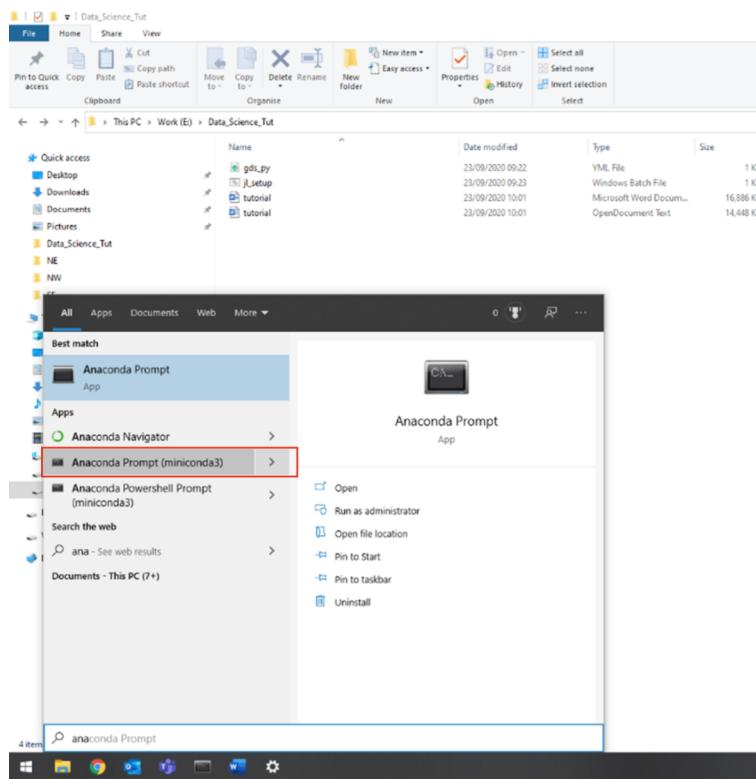


- Untick all boxes in the window (unless you want further information on Mini-conda, which will open in your browser) and click ***Finish***.

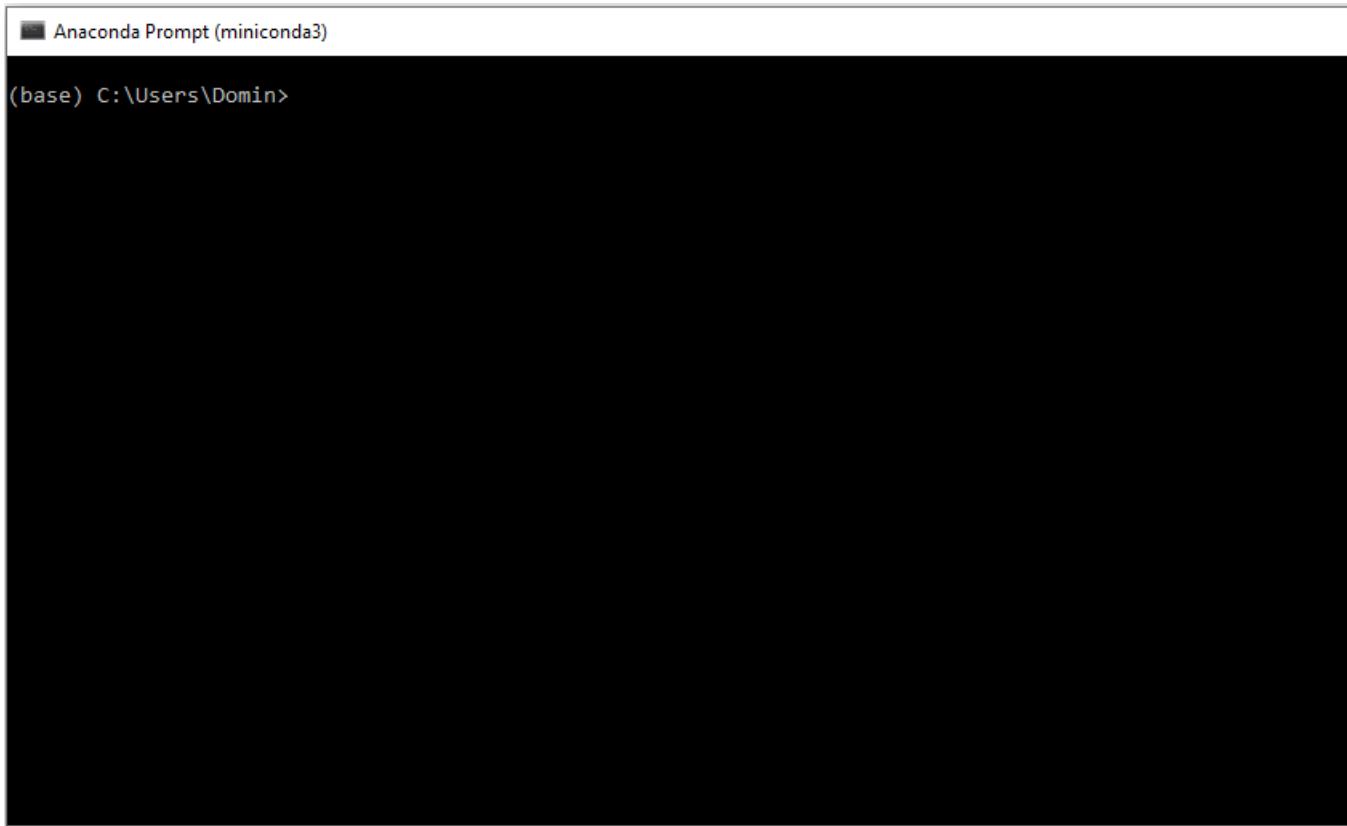
## Running Minicoda



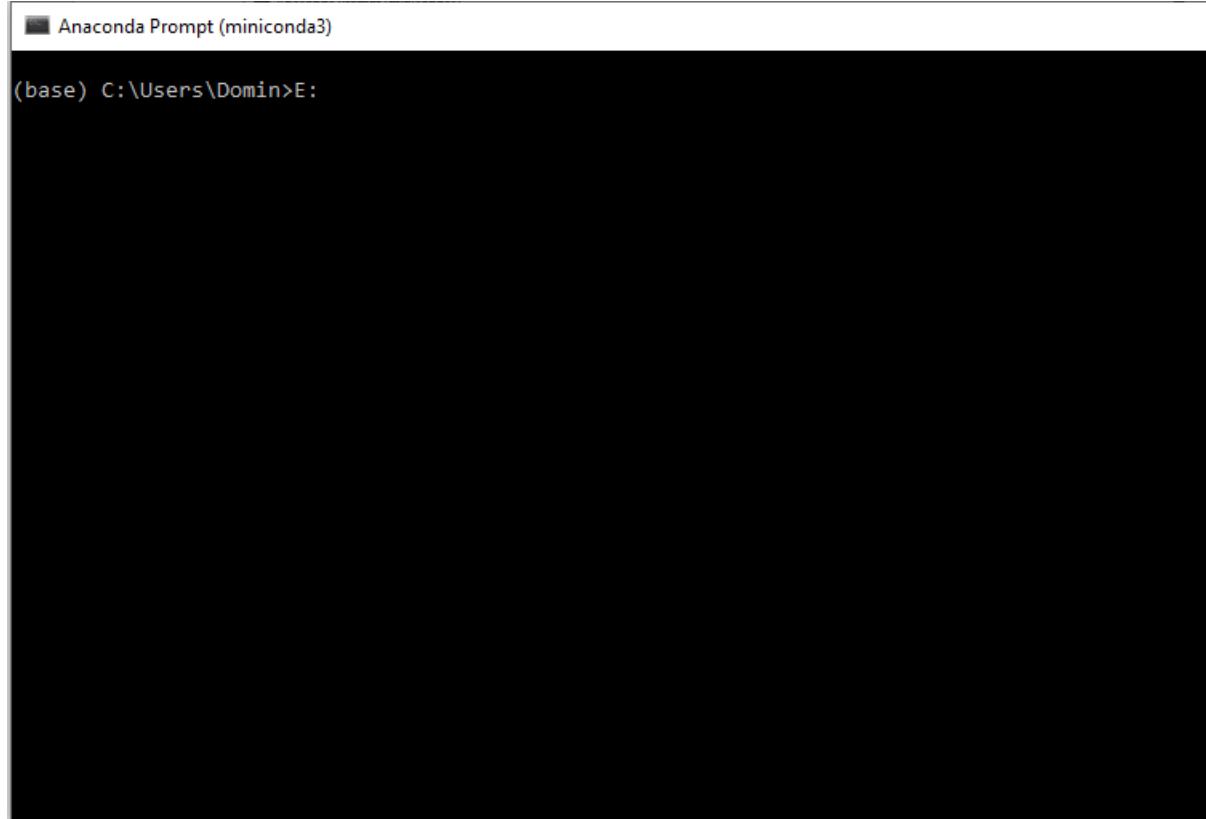
- Open Miniconda by clicking on the Windows icon on the bottom left of your screen and either type **Anaconda** or look for the **Anaconda** folder in the menu.



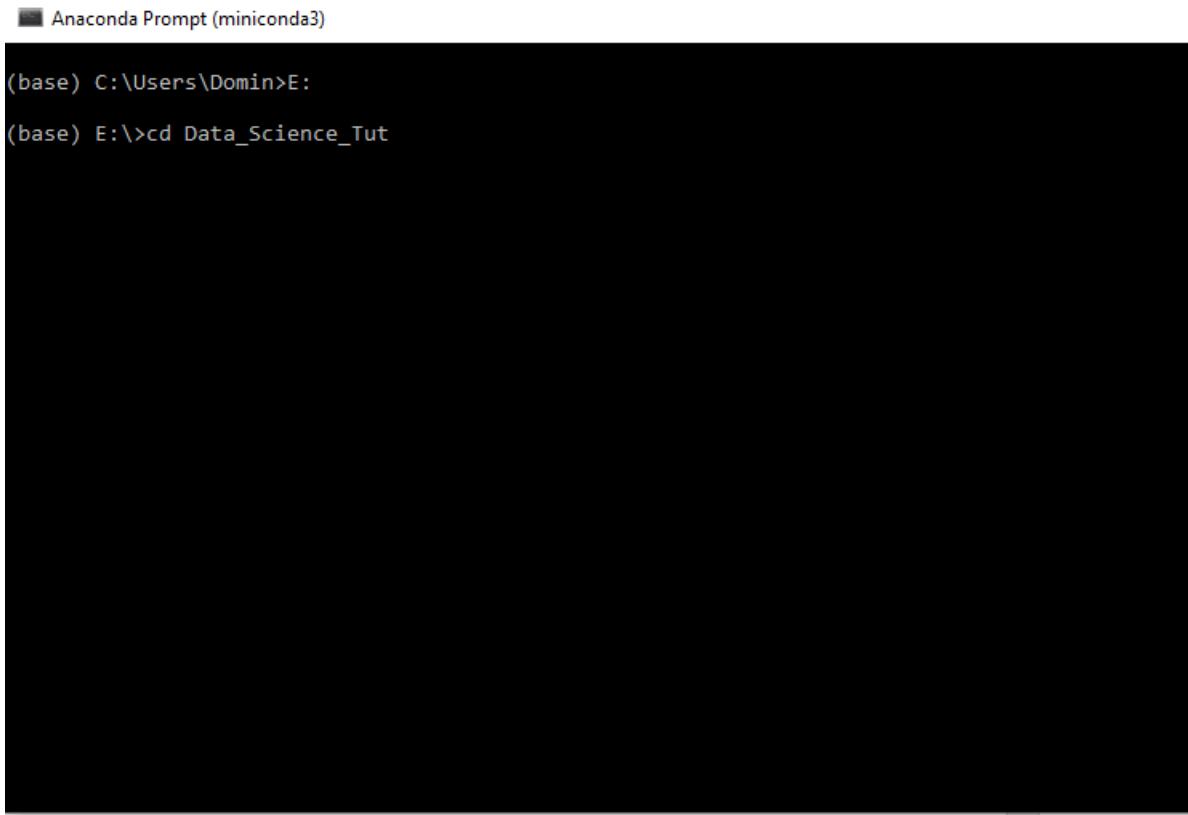
- To open Miniconda, click on **Anaconda Prompt (miniconda3)**. Note: From now on we will refer to the prompt as **Anaconda Prompt**



- This will open the Anaconda command prompt.



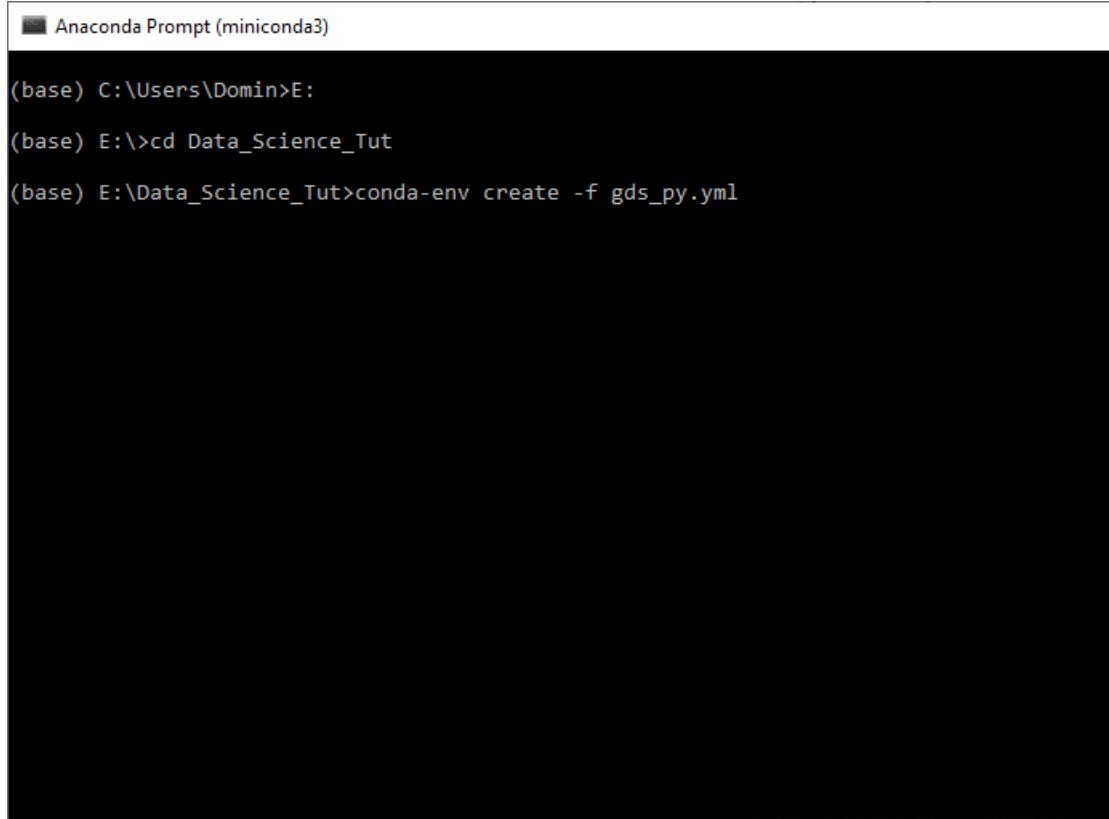
---



The screenshot shows a terminal window titled "Anaconda Prompt (miniconda3)". The command history is displayed:

```
(base) C:\Users\Domin>E:  
(base) E:>\cd Data_Science_Tut
```

- You now need to navigate to the folder that contains your environment (`_gds_py.yml_`) and setup (`_jl_setup.bat_`) files.
- you can move to the folder by running `cd` to move forward through folders and `cd ..` to move backwards.
- To run a command you simply press enter.
- If your files are stored in e.g. C:/Users/Domin/Desktop/GDS\_2020 you would write `cd Desktop/GDS_2020`
- If your files are stored in a different location e.g. E:/Data\_science\_Tut, you would run `E:` (to switch the harddrive) followed by `cd Data_science_Tut`.

A screenshot of an Anaconda Prompt window titled "Anaconda Prompt (miniconda3)". The window shows a command-line interface with the following text:

```
(base) C:\Users\Domin>E:  
(base) E:>\>cd Data_Science_Tut  
(base) E:\Data_Science_Tut>conda-env create -f gds_py.yml
```

The prompt shows the user navigating to a directory and then running the command "conda-env create -f gds\_py.yml". The command is partially cut off at the end.

- Once you have navigated to the location of your files, write the following in the Anaconda prompt and press enter to run it.

```
conda-env create -f gds_py.yml
```

- This will install all packages that are required to complete the course and setup your Python environment. \*\* Note: This might take a while as it is downloading all packages (~ 500 MB).\*\*

---

```
■ Anaconda Prompt (miniconda3) - conda env create -f gds_py.yml

(base) C:\Users\Domin>E:
(base) E:>\cd Data_Science_Tut
(base) E:\Data_Science_Tut>conda env create -f gds_py.yml
EnvironmentSectionNotValid: The following section on 'E:\Data_Science_Tut\gds_py.yml' is invalid and will
- gds_env_version

Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
    current version: 4.8.3
    latest version: 4.8.5

Please update conda by running

$ conda update -n base -c defaults conda

Downloading and Extracting Packages
jupyter telemetry-0. | 8 KB    | #####=====
jupyter core-4.6.3   | 94 KB   | =====#
aws-checksums-0.1.9 | 51 KB   | =====#
glib-5.2.1          | 84 KB   | =====#
zeromq-4.3.2        | 8.8 MB  | =====#
```

- The packages that are being installed will be shown in the Anaconda prompt.

```

[Anaconda Prompt (miniconda3)]
dataShader-0.11.1      | 14.0 MB   | #####
libnetcdf-4.7.4        | 601 KB    | #####
wheel-0.35.1           | 29 KB     | #####
networkx-2.5            | 1.2 MB    | #####
bottleneck-1.3.2       | 115 KB    | #####
nbclient-0.5.0          | 60 KB     | #####
entrypoints-0.3         | 12 KB     | #####
pycosat-0.6.3           | 102 KB    | #####
tifffile-2020.9.3       | 118 KB    | #####
numpy-1.19.1             | 4.8 MB    | #####
gdal-3.1.2               | 1.3 MB    | #####
m2w64-gcc-libs-5.3.0     | 518 KB    | #####
aws-c-common-0.4.57       | 150 KB    | #####
pycodestyle-2.6.0         | 38 KB     | #####
msgpack-python-1.0.0       | 80 KB     | #####
Preparing transaction: done
Verifying transaction: done
Executing transaction: / b'Enabling notebook extension jupyter-js-widgets/extension...\n
done
#
# To activate this environment, use
#
#     $ conda activate gds
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) E:\Data_Science_Tut>

```

- Once all packages have been installed and your environment is created, you can activate the environment with the following command:

```
conda activate gds
```

### 0.0.1 Complete environment Setup

We will now complete the setup by installing the user interface (Jupyter Lab) which you will need to code.

---

```
■ Anaconda Prompt (miniconda3)

(base) C:\Users\Domin>E:
(base) E:\>cd Data_Science_Tut
(base) E:\Data_Science_Tut>conda activate gds
(gds) E:\Data_Science_Tut>
```

- Activate the environment by running :

```
conda activate gds
```

- You can see that the start of the line has changed from **(base)** to **(gds)**.

```
[Anaconda Prompt (miniconda3)]
(gds) E:\Data_Science_Tut>jl_setup.bat
(gds) E:\Data_Science_Tut>REM Run this once the gds environment is created and activated
(gds) E:\Data_Science_Tut>REM Add pip packages

(gds) E:\Data_Science_Tut>pip install -r https://github.com/darribas/gds_env/raw/master/gds_py
Collecting ablog==0.10.6
  Downloading ablog-0.10.6-py3-none-any.whl (48 kB)
    |██████████| 48 kB 971 kB/s
Collecting geoalchemy2==0.8.4
  Downloading GeoAlchemy2-0.8.4-py2.py3-none-any.whl (25 kB)
Collecting jupyter-book==0.7.3
  Downloading jupyter_book-0.7.3-py3-none-any.whl (39 kB)
Collecting keplergl==0.2.0
  Downloading keplergl-0.2.0.tar.gz (7.0 MB)
    |██████████| 7.0 MB 6.8 MB/s
Collecting pygeoda==0.0.3
  Downloading pygeoda-0.0.3-cp37-cp37m-win_amd64.whl (7.5 kB)
    |██████████| 7.5 kB 6.4 MB/s
Collecting pytest-cov==2.10.0
  Downloading pytest_cov-2.10.0-py2.py3-none-any.whl (19 kB)
Collecting pytest-tornasync==0.6.0.post2
  Downloading pytest_tornasync-0.6.0.post2-py3-none-any.whl (6.6 kB)
Collecting sphinx_press_theme==0.5.1
  Downloading sphinx_press_theme-0.5.1.tar.gz (57 kB)
    |██████████| 57 kB 2.4 MB/s
Collecting alabaster
  Downloading alabaster-0.7.12-py2.py3-none-any.whl (14 kB)
Collecting sphinx-autodapi
```

- In the same prompt, run the following command to complete the environment setup

`jl_setup.bat`

- The prompt will show you the further packages that are being installed.

**NOTE:** This might take a while depending on your internet connection (at least 10-15 minutes). **NOTE:** Do not close the Anaconda prompt yet as we will need it again.

### 0.0.2 Check Installation

To make sure that your installation was successful and all packages have been installed we need to run one more step.

The screenshot shows a web browser window with a table of contents for a tutorial on miniconda installation and setup. The table of contents includes sections for download environment, miniconda download, miniconda installation, and running Python. The 'Check Installation' section is highlighted in blue.

	1 Download Environment and Setup files	2 Download Miniconda	3 Miniconda installation and Setup	4 Run Python
1.1 Environment File	2.1 Find your system type	3.1 Installation	4.1 Environment	
1.2 Setup	2.2 Download Miniconda	3.2 Setup	4.2 Run Python	

### 3.4 Check Installation

- To make sure that your installation was successful and all packages have been installed run one more step.

- Download the following file [check\\_py\\_stack.ipynb](https://raw.githubusercontent.com/darrrias/gds_env/master/gds_py/check_py_stack.ipynb)
- Save the file to the same location

```
[root@dom ~]# jupyter nbconvert --to html check_py_stack.ipynb
[nbconvert] Converting notebook check_py_stack.ipynb to html
[nbconvert] Success!
```

- Go back to your Anaconda prompt (make sure your environment is activated `conda activate gds`) and enter the following command:

```
jupyter nbconvert --to html --execute check_py_stack.ipynb
```

- This will check if all packages are working properly and produce an output .html file called `check_py_stack.html`

```
[root@dom ~]# jupyter nbconvert --to html check_py_stack.ipynb
[nbconvert] Converting notebook check_py_stack.ipynb to html
[nbconvert] Success!
```

- Download the following file `check_py_stack` by right clicking on it and selecting ***Save link as***.
- Save the file to the same location as all other files.

```
■ Anaconda Prompt (miniconda3) - jupyter nbconvert --to html --execute check_py_stack.ipynb
[--pin-version-as InstallLabExtensionApp.pin]
[extra_args [extra_args ...]]
jupyter-labextension: error: unrecognized arguments: \
(gds) E:\Data_Science_Tut>jupyter lab build -y
[LabBuildApp] JupyterLab 2.1.3
[LabBuildApp] Building in C:\Users\Domin\miniconda3\envs\gds\share\jupyter\lab
[LabBuildApp] Building jupyterlab assets (build:prod:minimize)
\
(gds) E:\Data_Science_Tut>jupyter lab clean -y
[LabCleanApp] Cleaning C:\Users\Domin\miniconda3\envs\gds\share\jupyter\lab...
[LabCleanApp] Removing staging...
[LabCleanApp] Success!

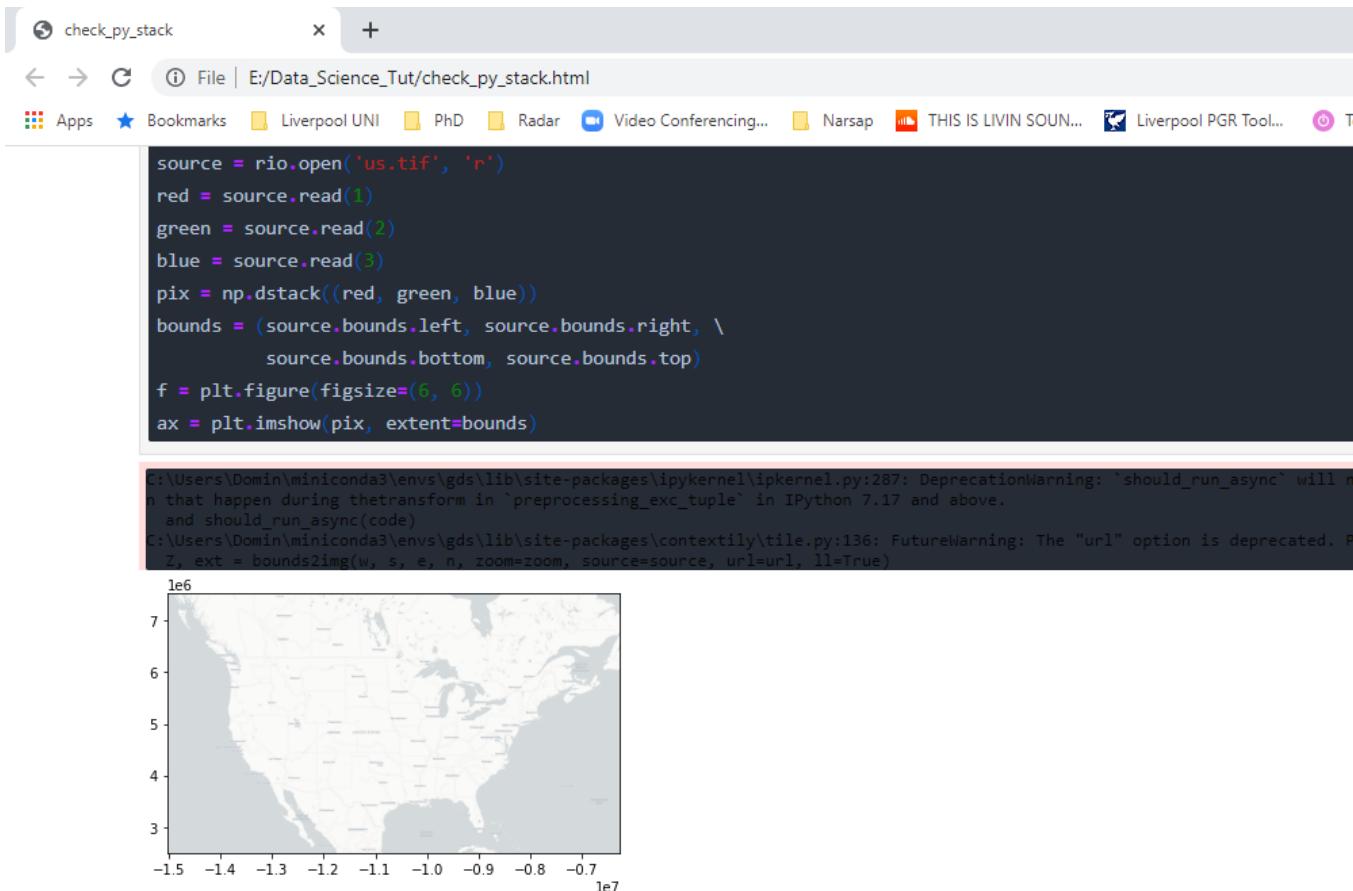
(gds) E:\Data_Science_Tut>conda clean --all -f -y
Remove all contents from the following package caches?
- C:\Users\Domin\AppData\Local\conda\conda\pkgs

(gds) E:\Data_Science_Tut>jupyter nbconvert --to html --execute check_py_stack.ipynb
[NbConvertApp] Converting notebook check_py_stack.ipynb to html
```

- Go back to your Anaconda prompt (**make sure your environment is activated** `conda activate gds`) and enter the following command:

```
jupyter nbconvert --to html --execute check_py_stack.ipynb
```

- This will check if all packages are working properly and produce an output .html file called `check_py_stack.html`

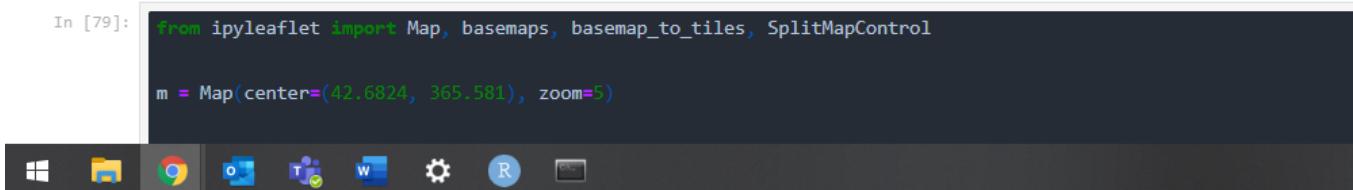


```
source = rio.open('us.tif', 'r')
red = source.read(1)
green = source.read(2)
blue = source.read(3)
pix = np.dstack((red, green, blue))
bounds = (source.bounds.left, source.bounds.right, \
           source.bounds.bottom, source.bounds.top)
f = plt.figure(figsize=(6, 6))
ax = plt.imshow(pix, extent=bounds)
```

C:\Users\Domin\miniconda3\envs\gds\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should\_run\_async` will no longer happen during the transform in `preprocessing\_exc\_tuple` in IPython 7.17 and above.  
and should\_run\_async(code)  
C:\Users\Domin\miniconda3\envs\gds\lib\site-packages\contextily\tile.py:136: FutureWarning: The "url" option is deprecated. Please use "url" instead.  
\_ext = bounds2img(w, s, e, n, zoom=zoom, source=source, url=url, ll=True)



```
In [78]: ax = db.plot()
ctx.add_basemap(ax, crs=db.crs.to_string())
```



```
In [79]: from ipyleaflet import Map, basemaps, basemap_to_tiles, SplitMapControl
```

```
m = Map(center=(42.6824, -75.581), zoom=5)
```

- Double clicking on the *check\_py\_stack.html* file will open the file in a browser and you can check if the code has produce an output in all cells.

## Running Python

Now that you have successfully installed Python/Apacheconda, you are ready to start coding. To launch your coding environment complete the following steps:

1. Start by opening an Anaconda Prompt (see first steps of section 3.2 on how to open an Anaconda prompt).
2. Navigate to the folder that you want to work in (It is recommended to have all your files in one folder with subfolders) using the `cd` command.
3. Activate your environment by running `conda activate gds`.

```
█ Select npm

(base) C:\Users\Domin>E:

(base) E:\>cd Data_Science_Tut

(base) E:\Data_Science_Tut>activate gds

(gds) E:\Data_Science_Tut>jupyter lab
[I 14:36:06.360 LabApp] Loading IPython parallel extension
[I 14:36:07.684 LabApp] JupyterLab extension loaded from C:\Users\Domin\miniconda3\envs\gds\lib\site-packages\jupyterlab\extension.py
[I 14:36:07.684 LabApp] JupyterLab application directory is C:\Users\Domin\miniconda3\envs\gds\app
[I 14:36:07.825 LabApp] [JupyterText Server Extension] Deriving a JupyterTextContentsManager from LabApp
[I 14:36:09.938 LabApp] Serving notebooks from local directory: E:\Data_Science_Tut
[I 14:36:09.938 LabApp] The Jupyter Notebook is running at:
[I 14:36:09.939 LabApp] http://localhost:8888/?token=92fc323ff8a5df8cc3f807871367a4a057d6bbe9f07890f
[I 14:36:09.939 LabApp] or http://127.0.0.1:8888/?token=92fc323ff8a5df8cc3f807871367a4a057d6bbe9f07890f
[I 14:36:09.939 LabApp] Use Control-C to stop this server and shut down all kernels (twice to quit)
[C 14:36:10.000 LabApp]

To access the notebook, open this file in a browser:
  file:///C:/Users/Domin/AppData/Roaming/jupyter/runtime/nbserver-21568-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=92fc323ff8a5df8cc3f807871367a4a057d6bbe9f07890f
  or http://127.0.0.1:8888/?token=92fc323ff8a5df8cc3f807871367a4a057d6bbe9f07890f
[I 14:36:14.795 LabApp] Build is up to date
[I 14:36:19.353 LabApp] Kernel started: 201bfef8-88bf-40e9-b02d-b7b5a23a85f7
[I 14:38:21.312 LabApp] Build is up to date
```

4. Run the command `jupyter lab` to start your coding interface. The coding interface will launch in your default browser (We recommend using Chrome or Firefox).

If your default browser is neither of the recommended, you can close the window that opens automatically, open Chrome/Firefox and past the URL from the Anaconda Prompt.

