



Installation of requirements

For this course, we require the installation of Python. We use Python using Anaconda distributions. We use Miniconda due to it is a lightweight distribution of Python. In this distribution, we install everything we need from zero.

Miniconda

To start, download Miniconda; as we mentioned in the introduction, the advantage of using Miniconda against the full Anaconda version is that this distribution is light; moreover, we have the advantage of installing just what we need instead of thousands of Python packages that we would not use.

To download, go to the following link:

<https://docs.conda.io/en/latest/miniconda.html>

Inside the previous link, you could see first the Windows installers, download the corresponding installer with your version, either 64 or 32 bits

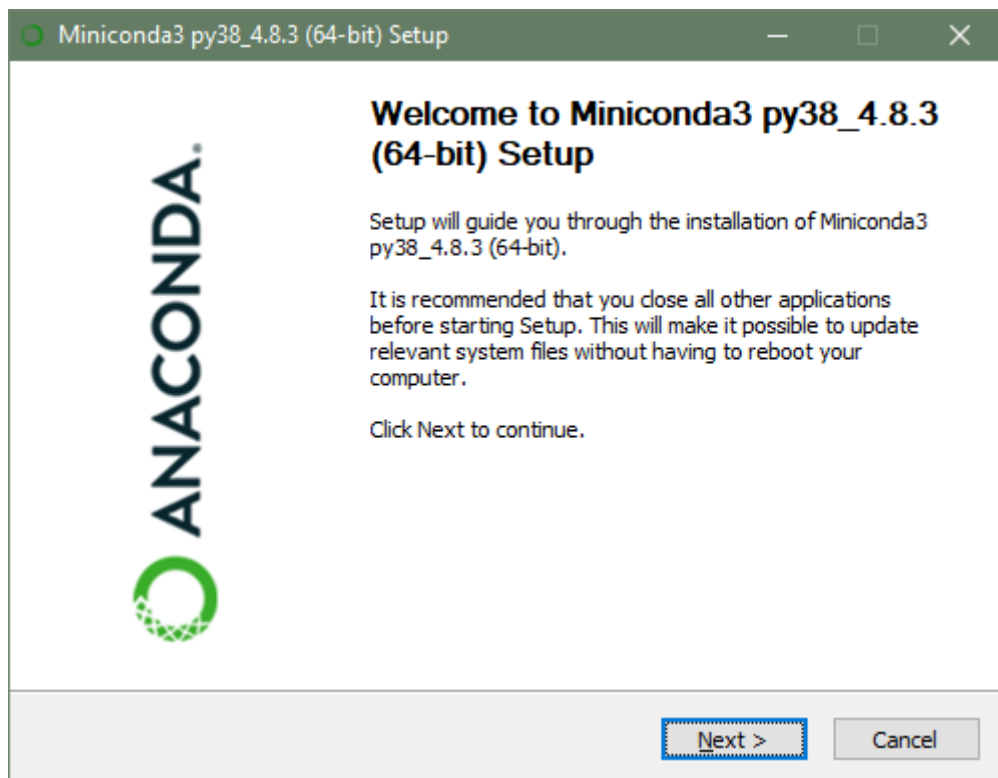
The screenshot shows the Miniconda documentation page. The left sidebar contains a search bar and navigation links: Conda, Conda-build, Miniconda (selected), Windows installers, MacOSX installers, Linux installers, Installing, Other resources, Help and support, Contributing, and Conda license. The main content area is titled 'Miniconda' and describes it as a free minimal installer for conda. It includes a 'conda install' command to install additional packages. Below this, the 'Windows installers' section features a table with columns for Python version, Name, Size, and SHA256 hash. The table lists installers for Python 3.8 and Python 2.7, each with 64-bit and 32-bit versions. The 32-bit installers for Python 3.8 are highlighted with a red box. Below the table, the 'MacOSX installers' section is partially visible.

Python version	Name	Size	SHA256 hash
Python 3.8	Miniconda3 Windows 64-bit	55.7 MiB	1f4ff67f051c815b600f144f5c4c3092af2005301d248b56281c36c1f4333e5
	Miniconda3 Windows 32-bit	49.6 MiB	415920293ae005a17afae4c275bd9180b6c07d8adf5e0cb9c99f0f090f976
Python 2.7	Miniconda2 Windows 64-bit	54.1 MiB	6973025404832944e074bf02bd88c4594908eeed4707051baa8f0dba40f326c
	Miniconda2 Windows 32-bit	47.7 MiB	c8049d26f6b6954057bcd4e99ad72d1ffa13f4aeb218e64e641504437b2617b

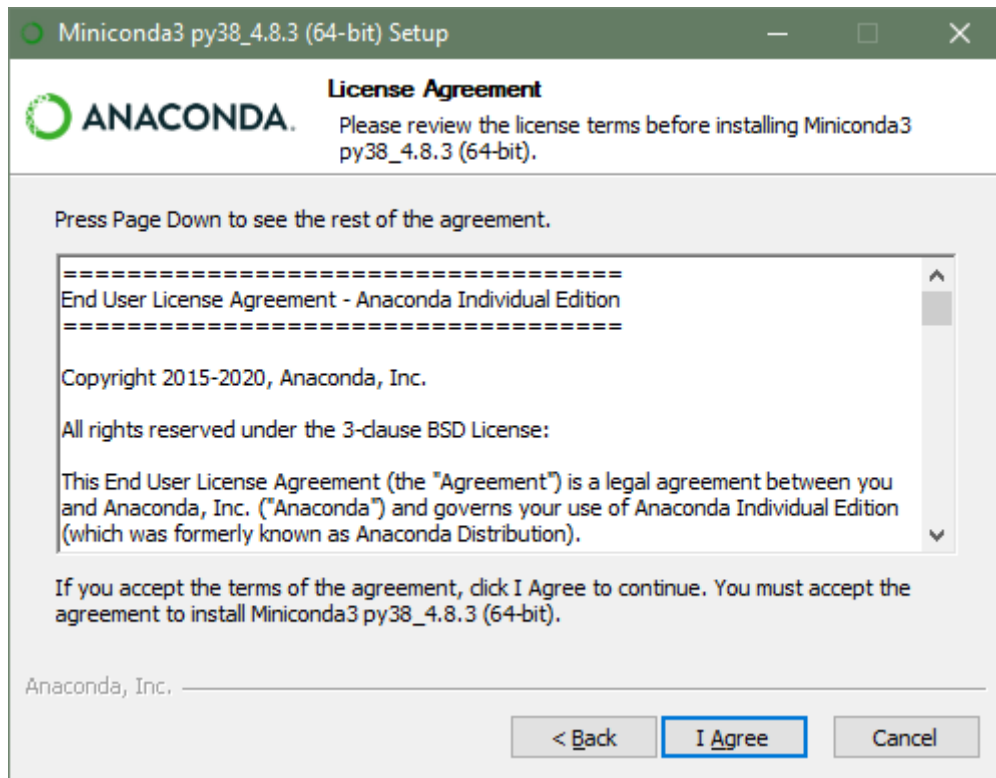
Once you download the installer you would have the following file:

Miniconda3-latest-Windows-x86_64.exe

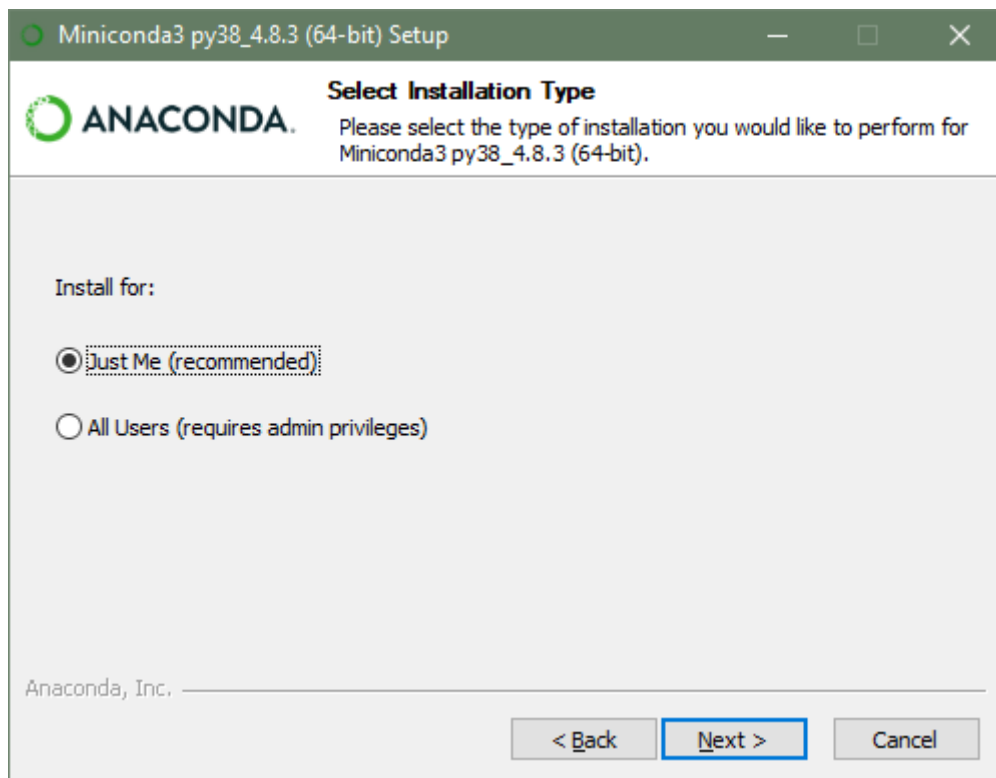
Open the installer with double click, click on "Next" to start the installation



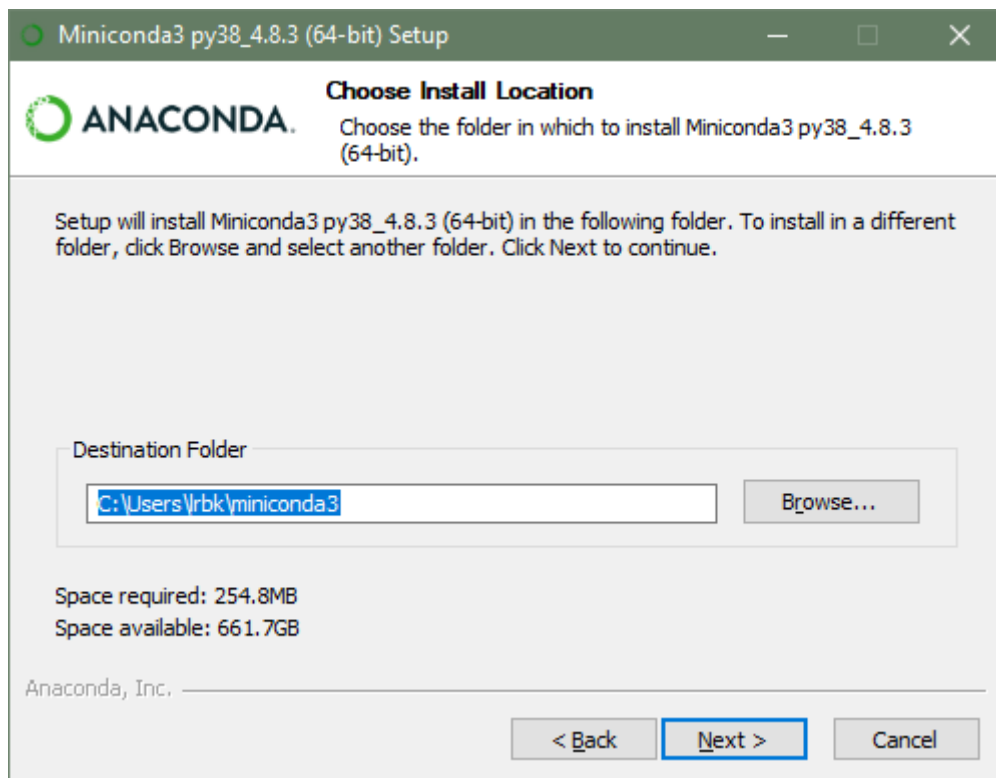
Click on "Agree" to accept the therms and conditions



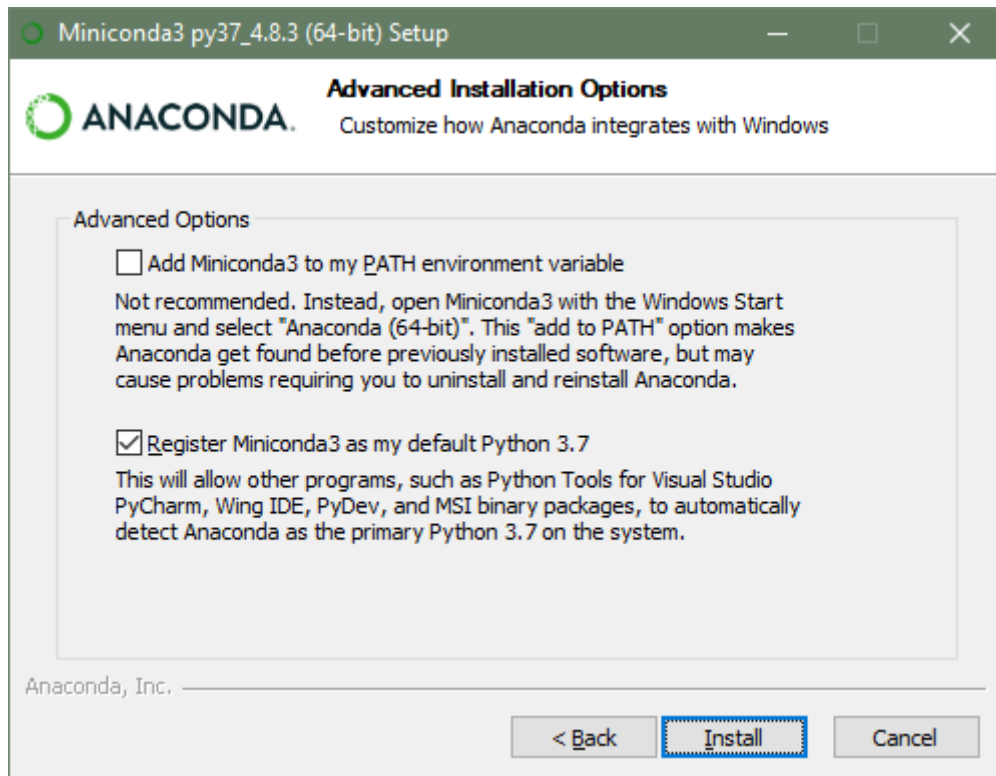
Leave by default the kind of installation and click on "Next."



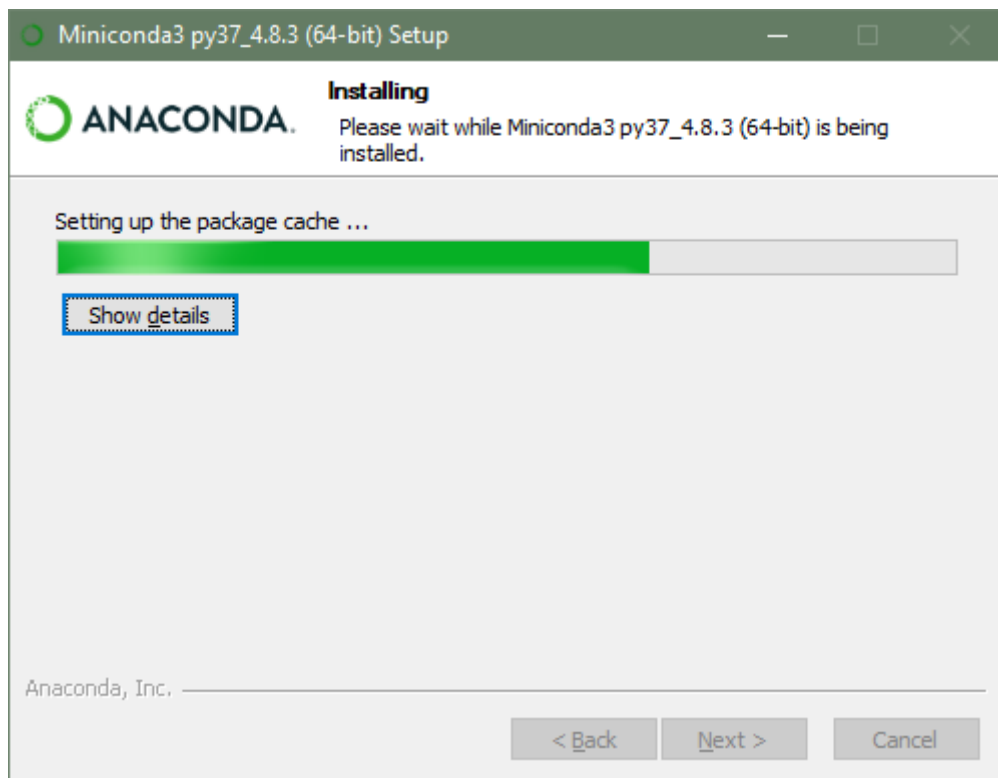
Leave as default the installation path and click on "Next"



Finally, click on "Install"



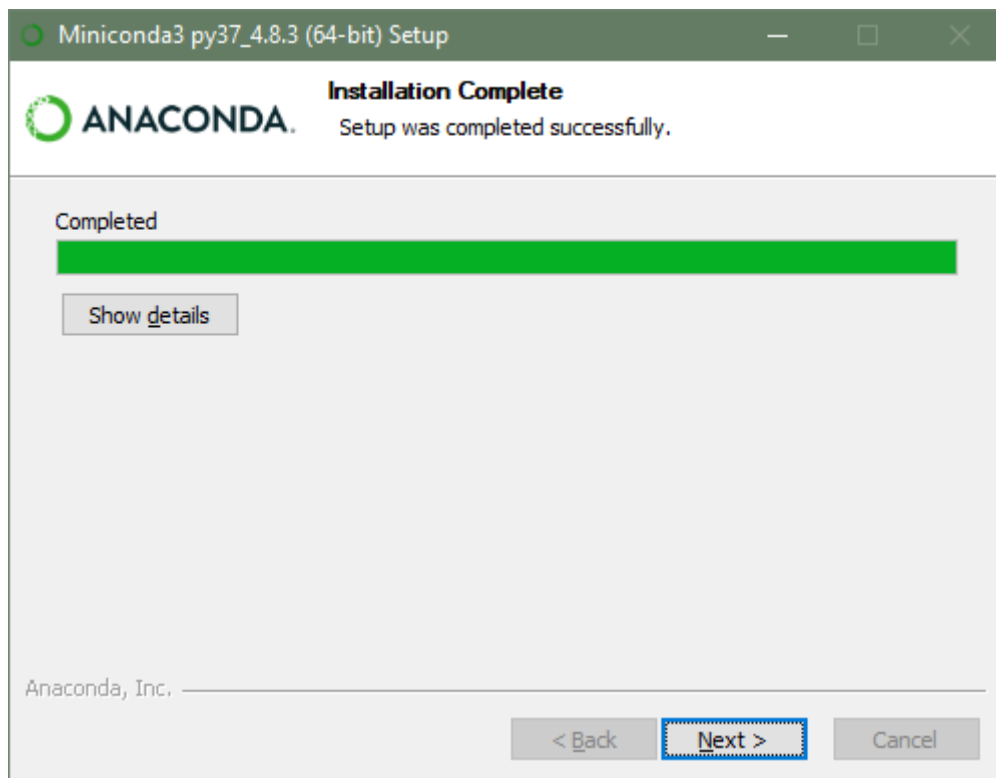
The installation can take some minutes to finish



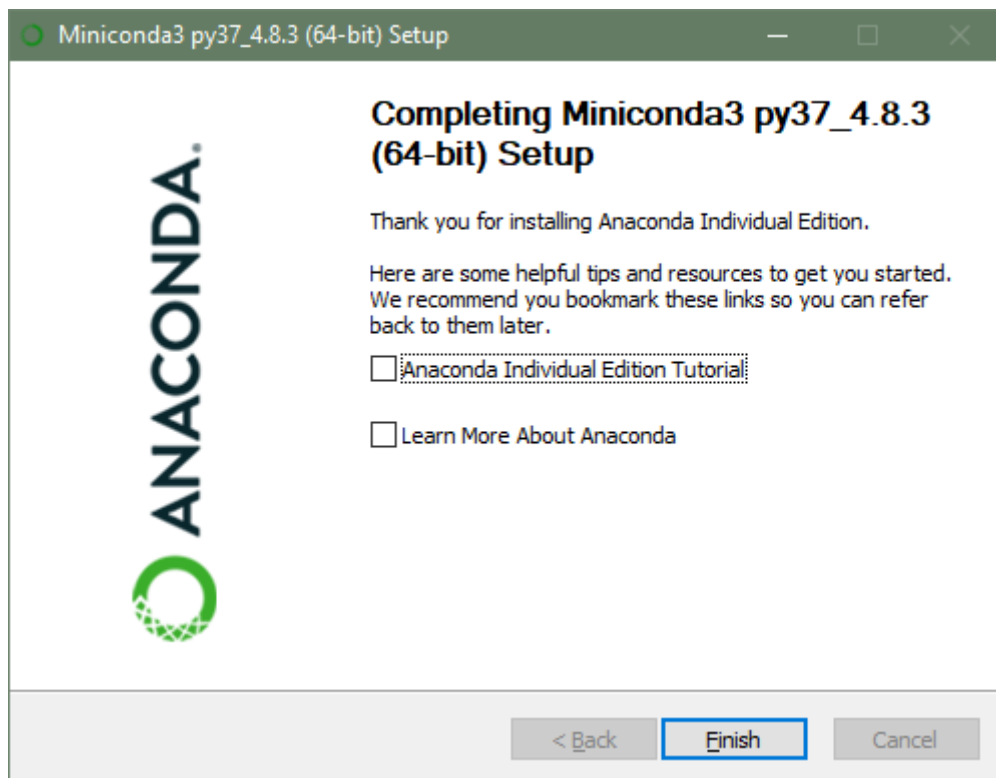
Once the installation is complete, click on "Next"



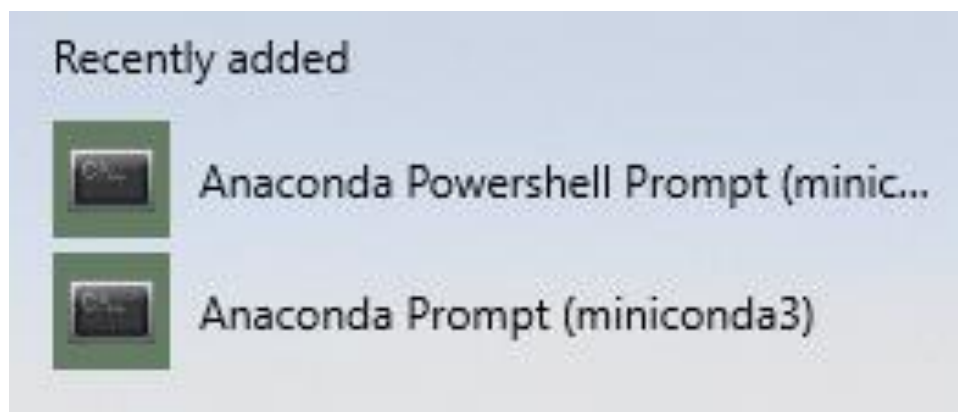
| Sustainable water management



To finish the installation process, click on "Finish"



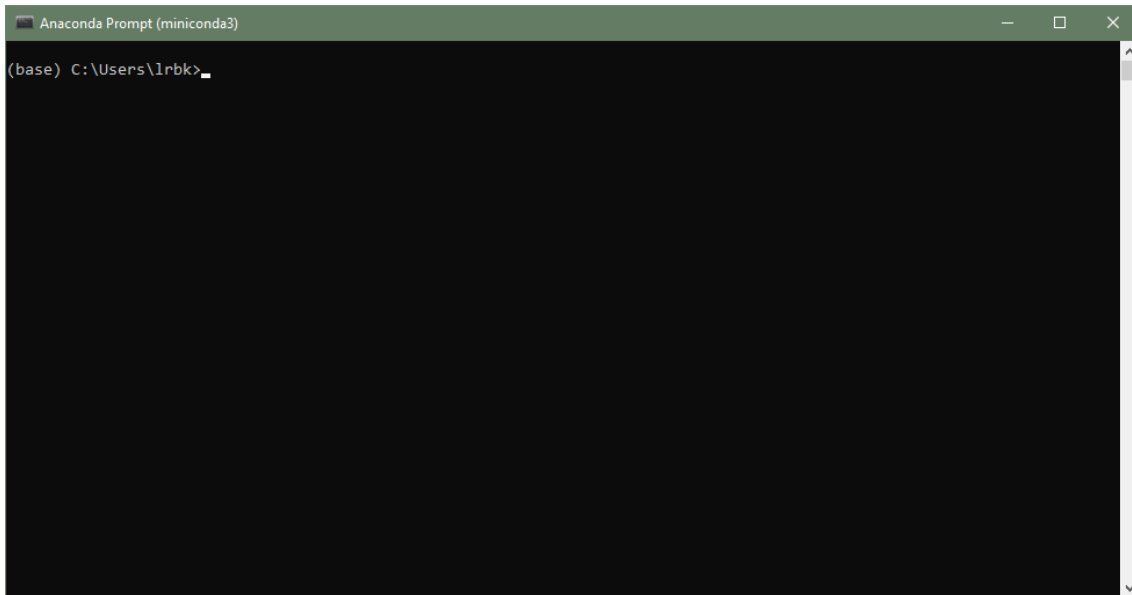
To open Miniconda, press the “Windows” button in your keyboard, and you could see that Miniconda is added to the Recently added part



Open the one that says “**Anaconda Prompt (miniconda3)**”, a window like the following will prompt.



|Sustainable water management

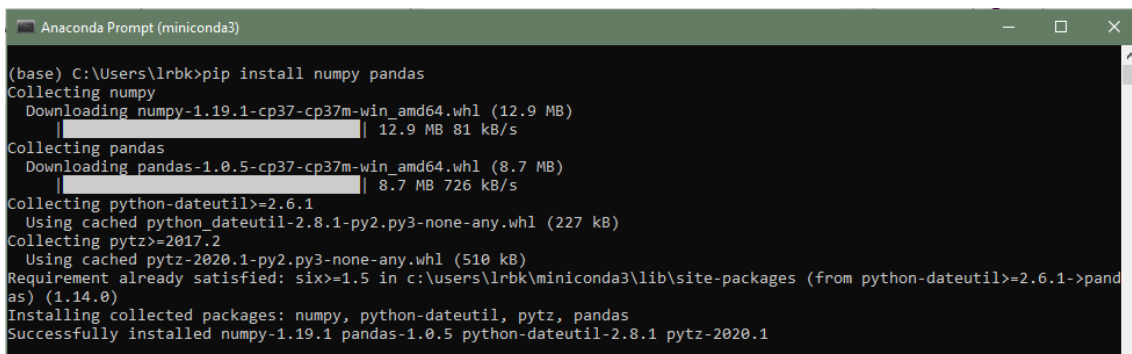


Now that we have Python3 installed install the libraries we want or that are going to be useful

First, install "numpy" and "pandas"

```
pip install numpy pandas
```

The installation process finishes by its own



Finally, we add the libraries "matplotlib" and "scipy"



```
pip install matplotlib scipy
```

Let's install "Jupyter Lab," an interface that allows an easy edition of our scripts, the installation is made through "conda," which is the tool for packages management of Miniconda

```
conda install -c conda-forge jupyterlab
```

It may ask you if you want to install a list of packages, write "y" and press "enter."

```
Select Anaconda Prompt (miniconda3) - conda install -c conda-forge jupyterlab

pygments          conda-forge/noarch::pygments-2.6.1-py_0
pyparsing          conda-forge/noarch::pyparsing-2.4.7-pyh9f0ad1d_0
pysistent          conda-forge/win-64::pysistent-0.16.0-py37h8055547_0
python-dateutil    conda-forge/noarch::python-dateutil-2.8.1-py_0
python_abi         conda-forge/win-64::python_abi-3.7-1_cp37m
pywinpty           conda-forge/win-64::pywinpty-0.5.7-py37_0
pyzmq              conda-forge/win-64::pyzmq-19.0.1-py37h453f00a_0
send2trash         conda-forge/noarch::send2trash-1.5.0-py_0
terminado          conda-forge/win-64::terminado-0.8.3-py37hc8dfbb8_1
testpath           conda-forge/noarch::testpath-0.4.4-py_0
tornado            conda-forge/win-64::tornado-6.0.4-py37hfa6e2cd_0
traitlets          conda-forge/win-64::traitlets-4.3.3-py37hc8dfbb8_1
wcwidth            conda-forge/noarch::wcwidth-0.2.5-pyh9f0ad1d_1
webencodings       conda-forge/noarch::webencodings-0.5.1-py_1
winpty             conda-forge/win-64::winpty-0.4.3-4
zeromq             conda-forge/win-64::zeromq-4.3.2-ha925a31_3
zipp               conda-forge/noarch::zipp-3.1.0-py_0

The following packages will be UPDATED:

ca-certificates    pkgs/main::ca-certificates-2020.1.1-0 --> conda-forge::ca-certificates-2020.6.20-hecda079_0
certifi            pkgs/main::certifi-2020.4.5.1-py37_0 --> conda-forge::certifi-2020.6.20-py37hc8dfbb8_0
conda              pkgs/main::conda-4.8.3-py37_0 --> conda-forge::conda-4.8.3-py37hc8dfbb8_1

The following packages will be SUPERSEDED by a higher-priority channel:

openssl            pkgs/main --> conda-forge

Proceed ([y]/n)?
```

Once the installation of "Jupyter Lab" finishes, you could access it bt typing "Jupyter lab."

```
jupyter lab
```



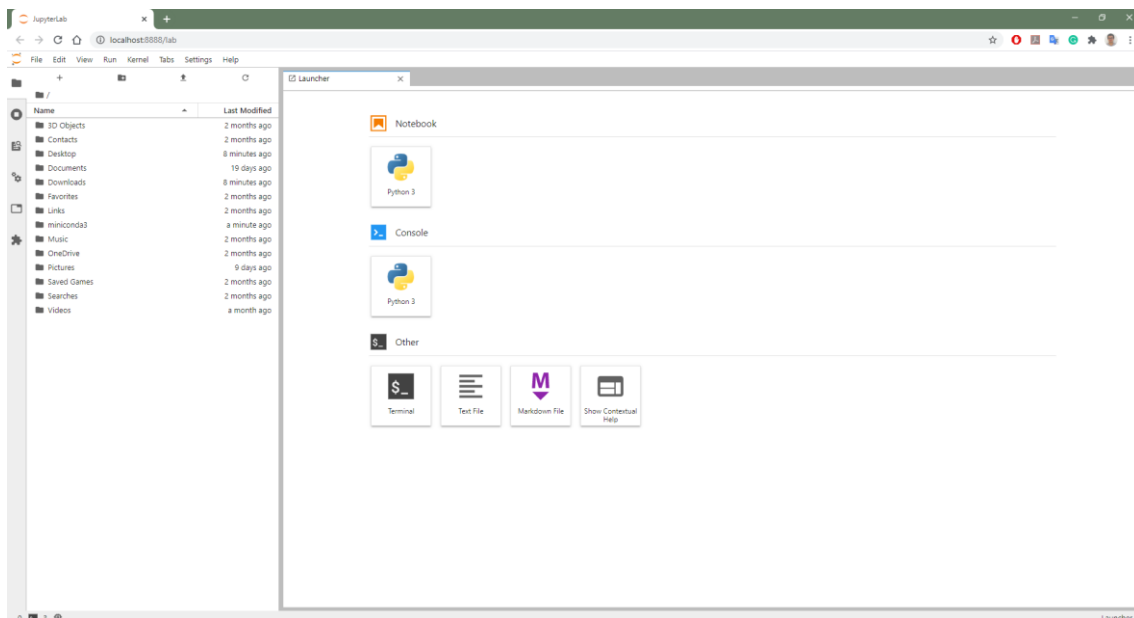
|Sustainable water management

```
Anaconda Prompt (miniconda3) - jupyter lab

(base) C:\Users\lrbk>jupyter lab
[I 16:56:57.843 LabApp] JupyterLab extension loaded from C:\Users\lrbk\miniconda3\lib\site-packages\jupyterlab
[I 16:56:57.843 LabApp] JupyterLab application directory is C:\Users\lrbk\miniconda3\share\jupyter\lab
[I 16:56:57.847 LabApp] Serving notebooks from local directory: C:\Users\lrbk
[I 16:56:57.847 LabApp] The Jupyter Notebook is running at:
[I 16:56:57.848 LabApp] http://localhost:8888/?token=0077b659b994f82dd6ec676597b38de0d4afc7f1619676a8
[I 16:56:57.848 LabApp] or http://127.0.0.1:8888/?token=0077b659b994f82dd6ec676597b38de0d4afc7f1619676a8
[I 16:56:57.848 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:56:57.959 LabApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/lrbk/AppData/Roaming/jupyter/runtime/nbserver-13572-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=0077b659b994f82dd6ec676597b38de0d4afc7f1619676a8
    or http://127.0.0.1:8888/?token=0077b659b994f82dd6ec676597b38de0d4afc7f1619676a8
```

You could see the interface of “Jupyter Lab.”



Since this point, you are ready to start with Session 1.

Before finishing the installation guide, we are going to install “Widgets”, which are extensions for Jupyter Lab. Open the “Anaconda Prompt” and run the next commands:

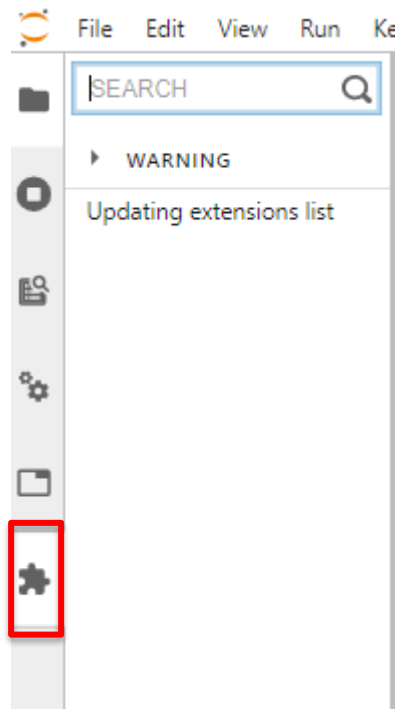


```
conda install -c conda-forge ipywidgets  
conda install -c conda-forge nodejs  
jupyter labextension install @jupyter-widgets/jupyterlab-manager
```

To enable the widgets, open "Jupyter Lab" from the Anaconda Prompt:

```
jupyter lab
```

Inside "Jupyter Lab" go to the extensions tab,



Clic on "Enable" and you could see the widgets "plugin"

If you cannot see the extensions tab, go to the "Settings" menu and clic on "Enable Extension Manager"

