



Adamson University | College of Engineering | Computer Engineering Department

Laboratory Guide

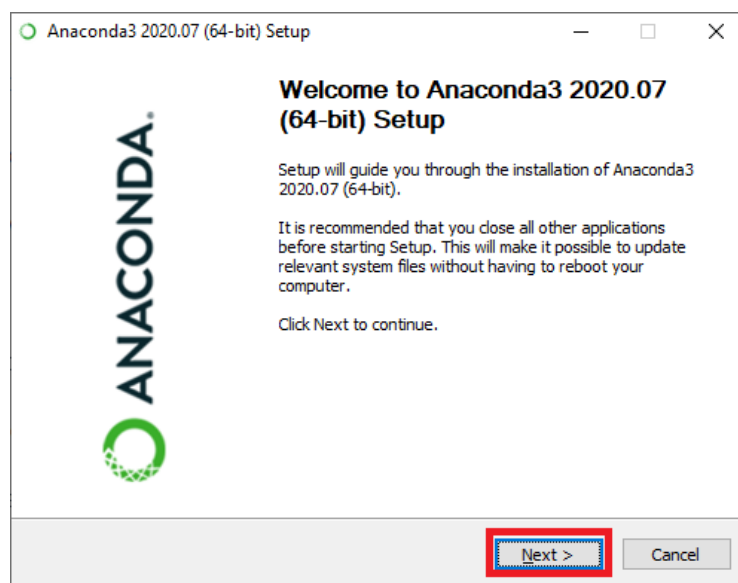
Linear Algebra

Setting up Python for Scientific programming

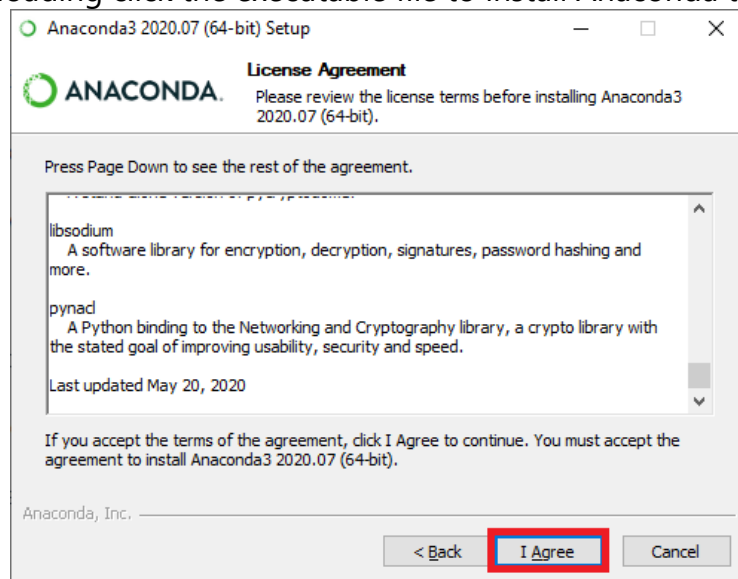
Installing Anaconda

We will be using Python for doing scientific and numerical programming for our course in Linear Algebra. Before you'd go ahead and download Python directly from their site, let me just stop you there since we will use a data science package to make future coding easier using Anaconda.

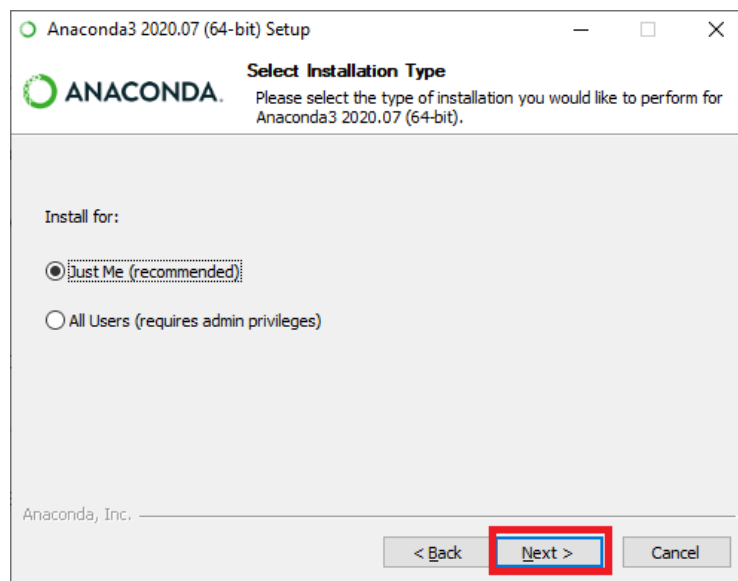
Anaconda is a data science package based on Python, that has built-in data science libraries for scientific and numerical computation. To download and install Anaconda, click here: <https://www.anaconda.com/products/individual>



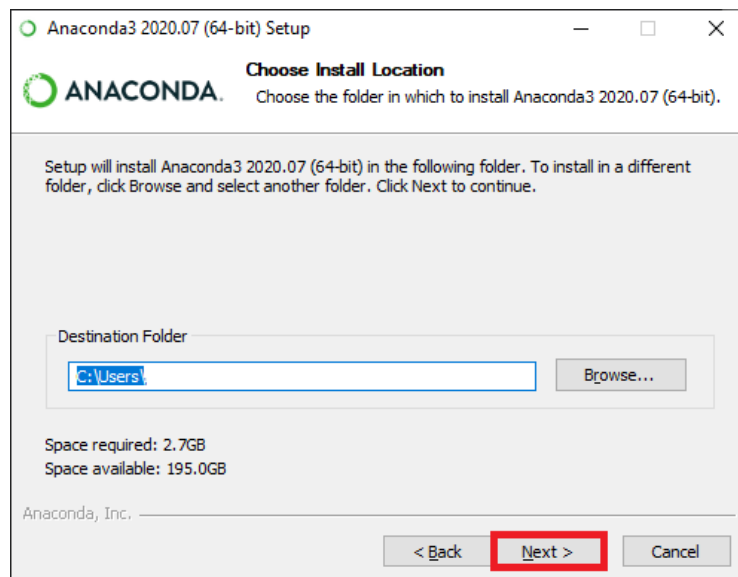
After downloading click the executable file to install Anaconda then proceed.



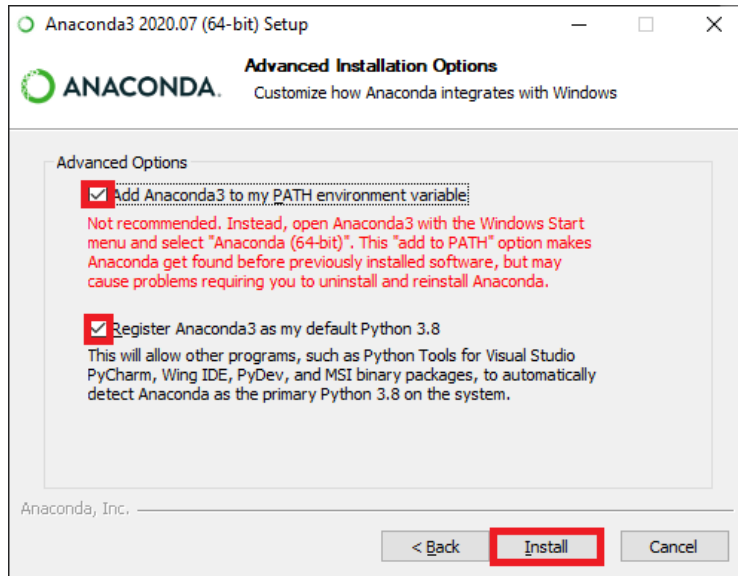
If you have the freedom and time to read the End-user license agreement, please do.
Click "I Agree" to proceed.



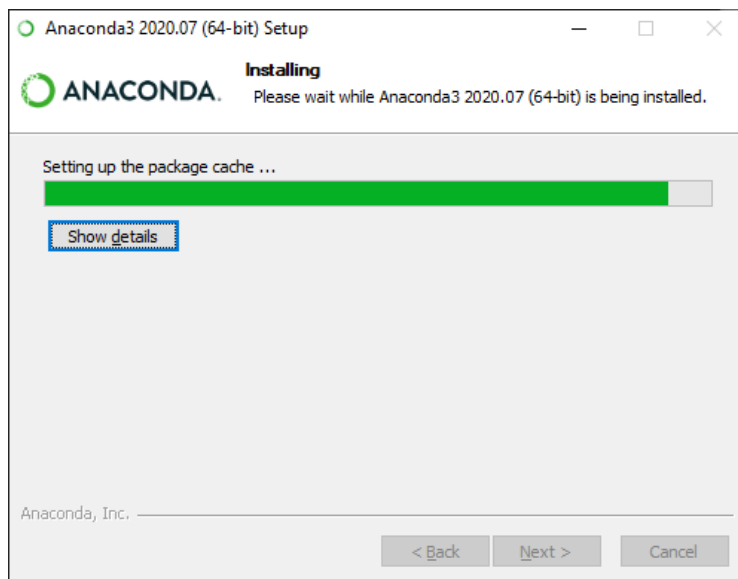
You can install it just for you or for all users, then proceed.



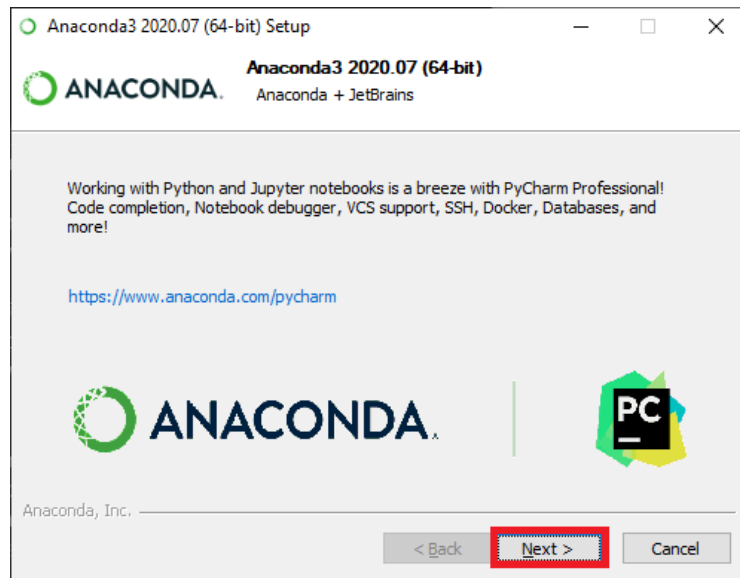
Choose the installation drive and folder to install Anaconda. If you have installed Anaconda before, you can just update it or uninstall it completely first.



Make sure to check both tick boxes. If you are doubtful about the first, don't worry you can trust me on this. We will be using the command line a lot to just check the first one for now. You can now then install Anaconda.

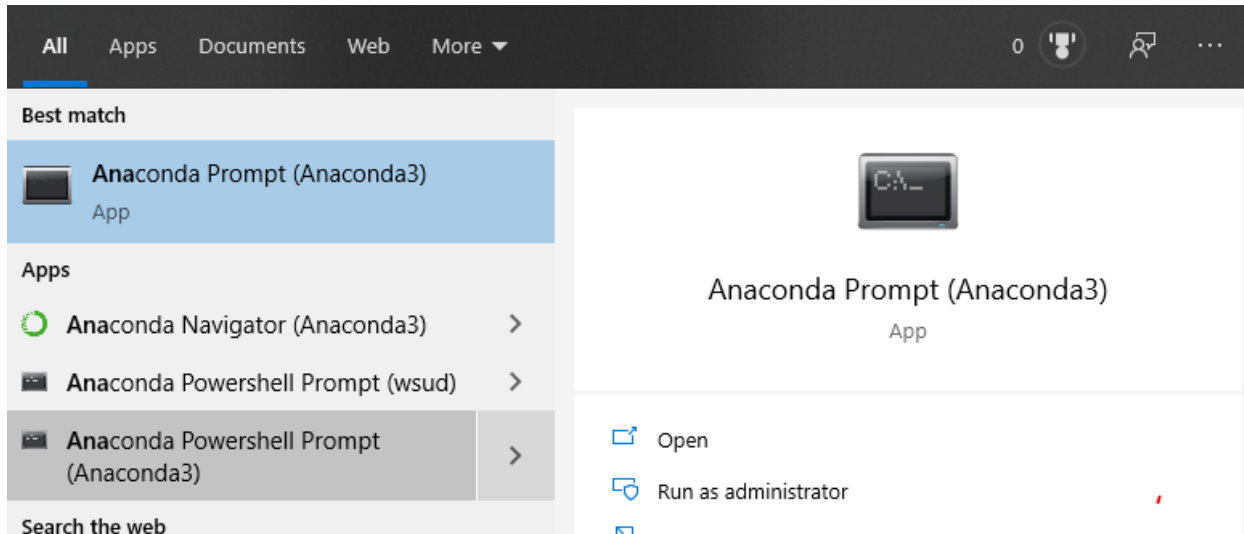


Just wait until the installation is complete. This will take a while depending on your computer specs.



Once everything's fine. Just click "Next" and finalize everything.

Testing Anaconda



Try to search Anaconda in your start menu and select the Anaconda Prompt



A command-line interface should appear. Make sure that there is a (base) indicator on the left side of the command line.

To check Python is working, type `python --version` to make sure python is installed.

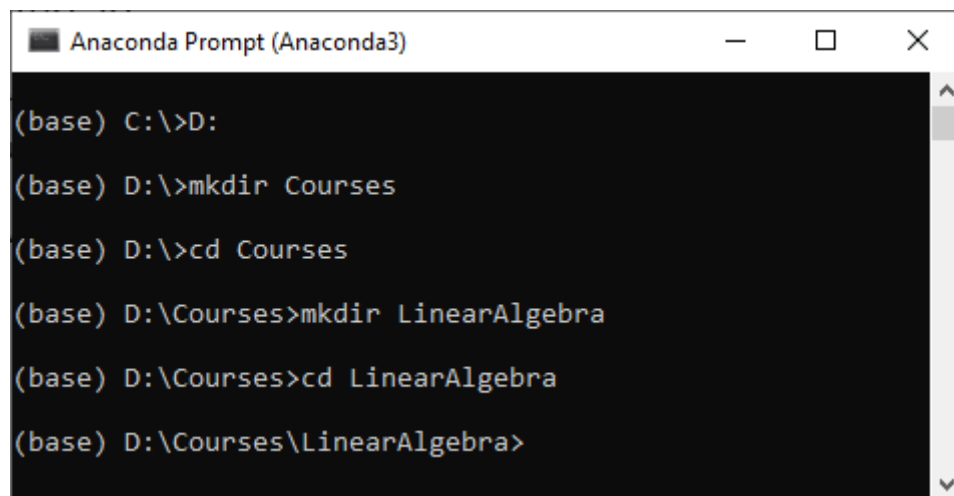


Trying JupyterLab

First, we need to organize our projects. Find a convenient place for all of your activities on your computer and create a folder named "LinearAlgebra".

For those who want to practice their command-line skills you can try to type the following:

> D:	<i>Go to your drive of choice</i>
> mkdir Courses	<i>This means make a directory named: "Courses"</i>
> cd Courses	<i>Go to the "Courses" directory</i>
> mkdir LinearAlgebra	<i>Make a directory named "LinearAlgebra"</i>
> cd LinearAlgebra	<i>Go to the "LinearAlgebra" folder</i>



```
Anaconda Prompt (Anaconda3)

(base) C:\>D:

(base) D:\>mkdir Courses

(base) D:\>cd Courses

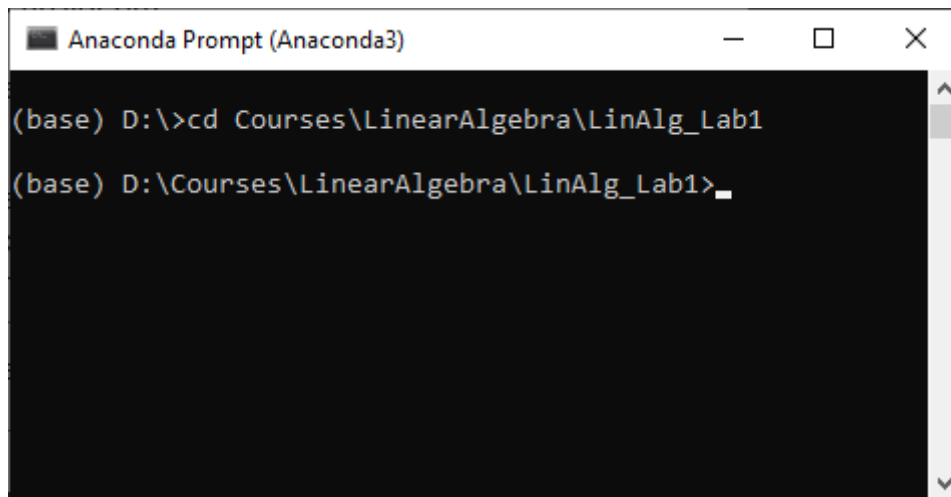
(base) D:\Courses>mkdir LinearAlgebra

(base) D:\Courses>cd LinearAlgebra

(base) D:\Courses\LinearAlgebra>
```

Now our project folder is ready we can now use this for our activities. In your LinearAlgebra folder create a new directory for your Laboratory Activity.

Open your Anaconda Prompt and access your First Laboratory Activity, mine is named as "LinAlg_Lab1"

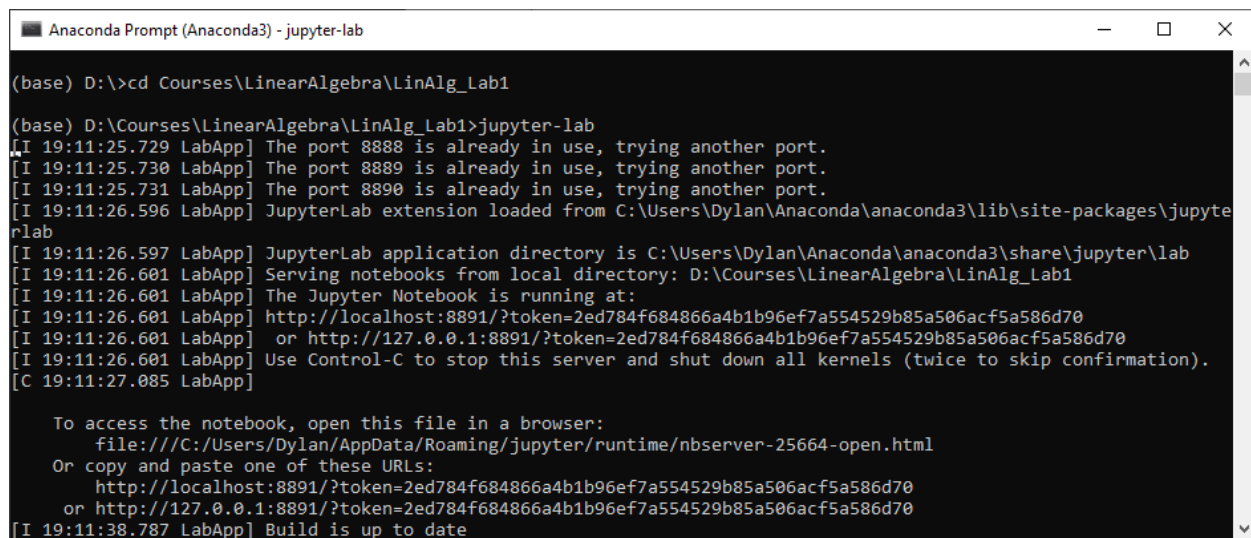


```
Anaconda Prompt (Anaconda3)

(base) D:\>cd Courses\LinearAlgebra\LinAlg_Lab1

(base) D:\Courses\LinearAlgebra\LinAlg_Lab1>
```

Once you're in your Lab Activity folder type `jupyter-lab`



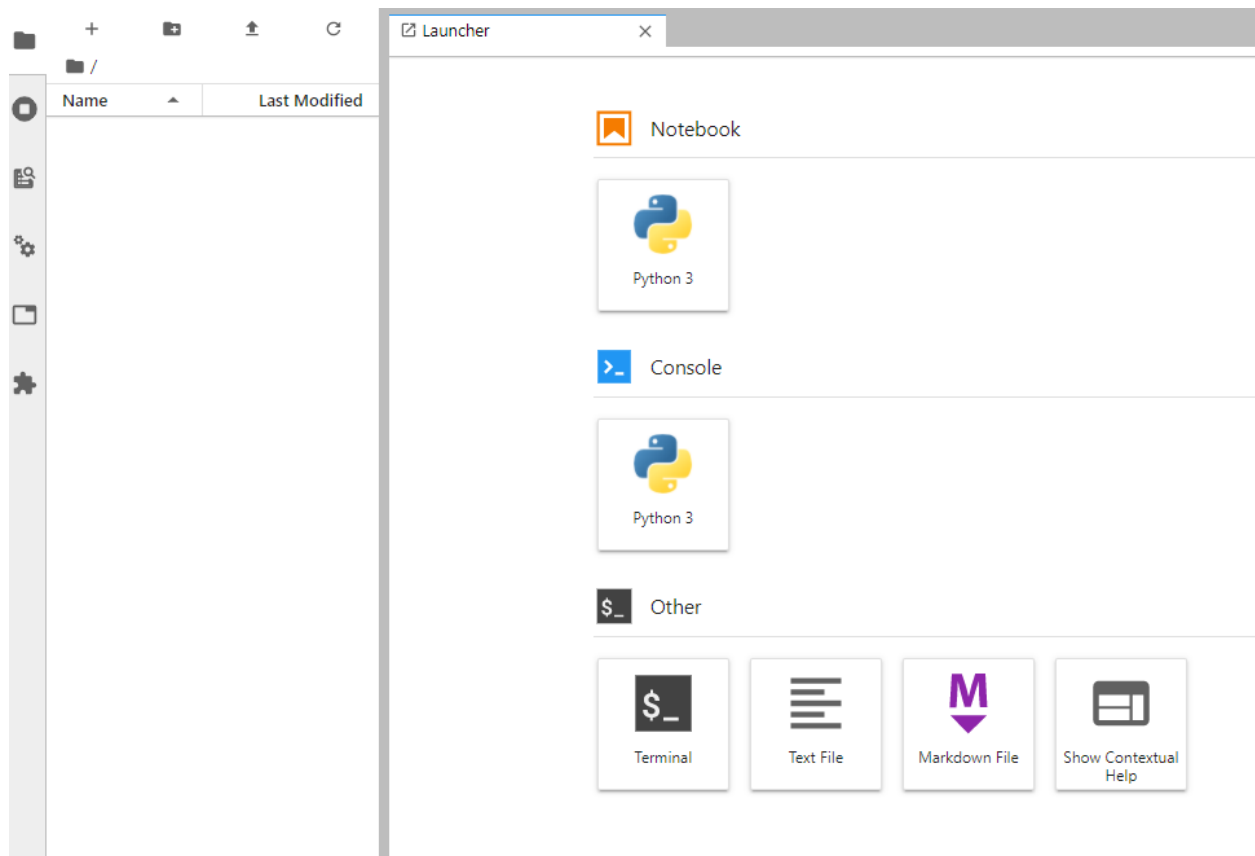
```
Anaconda Prompt (Anaconda3) - jupyter-lab

(base) D:\>cd Courses\LinearAlgebra\LinAlg_Lab1

(base) D:\Courses\LinearAlgebra\LinAlg_Lab1>jupyter-lab
[I 19:11:25.729 LabApp] The port 8888 is already in use, trying another port.
[I 19:11:25.730 LabApp] The port 8889 is already in use, trying another port.
[I 19:11:25.731 LabApp] The port 8890 is already in use, trying another port.
[I 19:11:26.596 LabApp] JupyterLab extension loaded from C:\Users\Dylan\Anaconda\anaconda3\lib\site-packages\jupyterlab
[I 19:11:26.597 LabApp] JupyterLab application directory is C:\Users\Dylan\Anaconda\anaconda3\share\jupyter\lab
[I 19:11:26.601 LabApp] Serving notebooks from local directory: D:\Courses\LinearAlgebra\LinAlg_Lab1
[I 19:11:26.601 LabApp] The Jupyter Notebook is running at:
[I 19:11:26.601 LabApp] http://localhost:8891/?token=2ed784f684866a4b1b96ef7a554529b85a506acf5a586d70
[I 19:11:26.601 LabApp] or http://127.0.0.1:8891/?token=2ed784f684866a4b1b96ef7a554529b85a506acf5a586d70
[I 19:11:26.601 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 19:11:27.085 LabApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/Dylan/AppData/Roaming/jupyter/runtime/nbserver-25664-open.html
Or copy and paste one of these URLs:
    http://localhost:8891/?token=2ed784f684866a4b1b96ef7a554529b85a506acf5a586d70
    or http://127.0.0.1:8891/?token=2ed784f684866a4b1b96ef7a554529b85a506acf5a586d70
[I 19:11:38.787 LabApp] Build is up to date
```

Once the command-line has been populated with these logs you should be directed to your Jupyter Lab page.



There! It's done you have successfully launched Jupyter Lab, you are now ready to code!