Python Tutorials

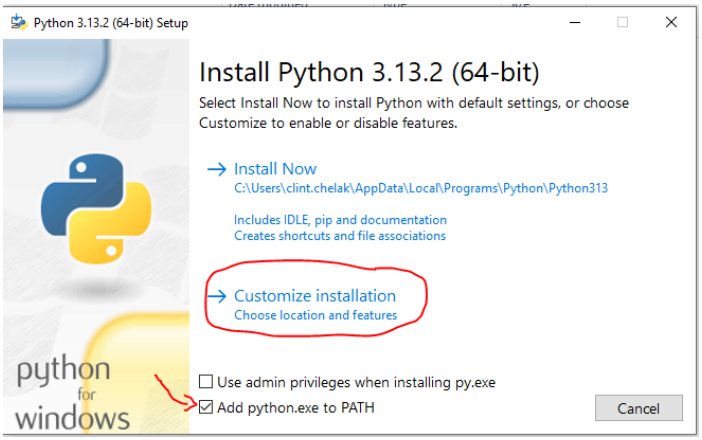
Table of Contents

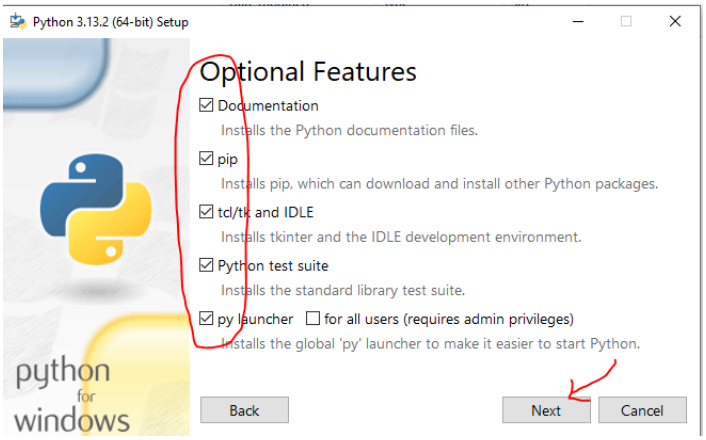
Intro

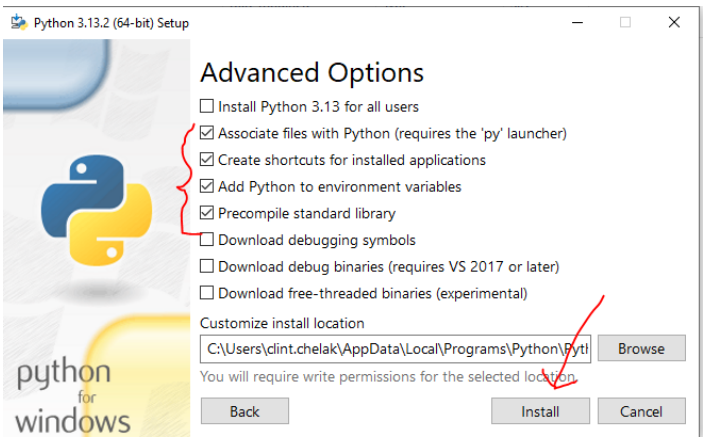
The following guides will be added to the lab notes in the future to ensure students can get up to speed on python. However, as is the nature with installation processes on a Windows machine, this tutorial could fall out of date rather quickly.

How to Install Python

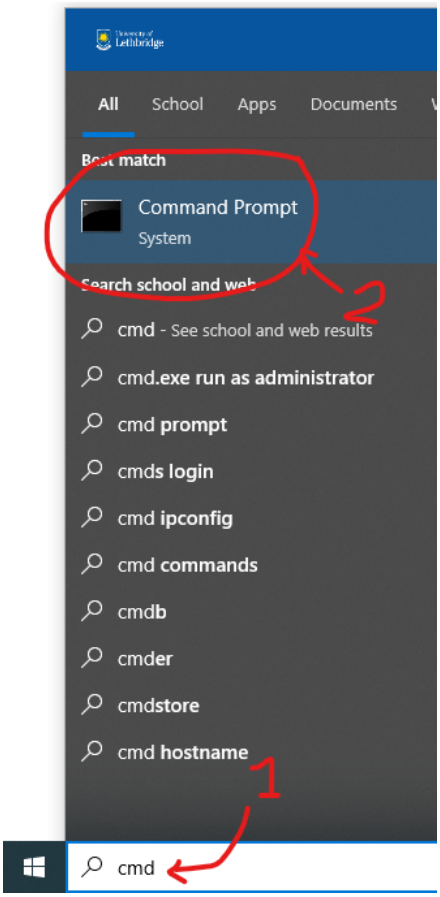
1. Install python onto the system, if it is not already. Download from the official website: https://www.python.org/downloads/
2. Follow the python install instructions (example below). I'd recommend using the custom installation and reading applying options that sound interesting and useful to you. The following is my recommended install for the class:



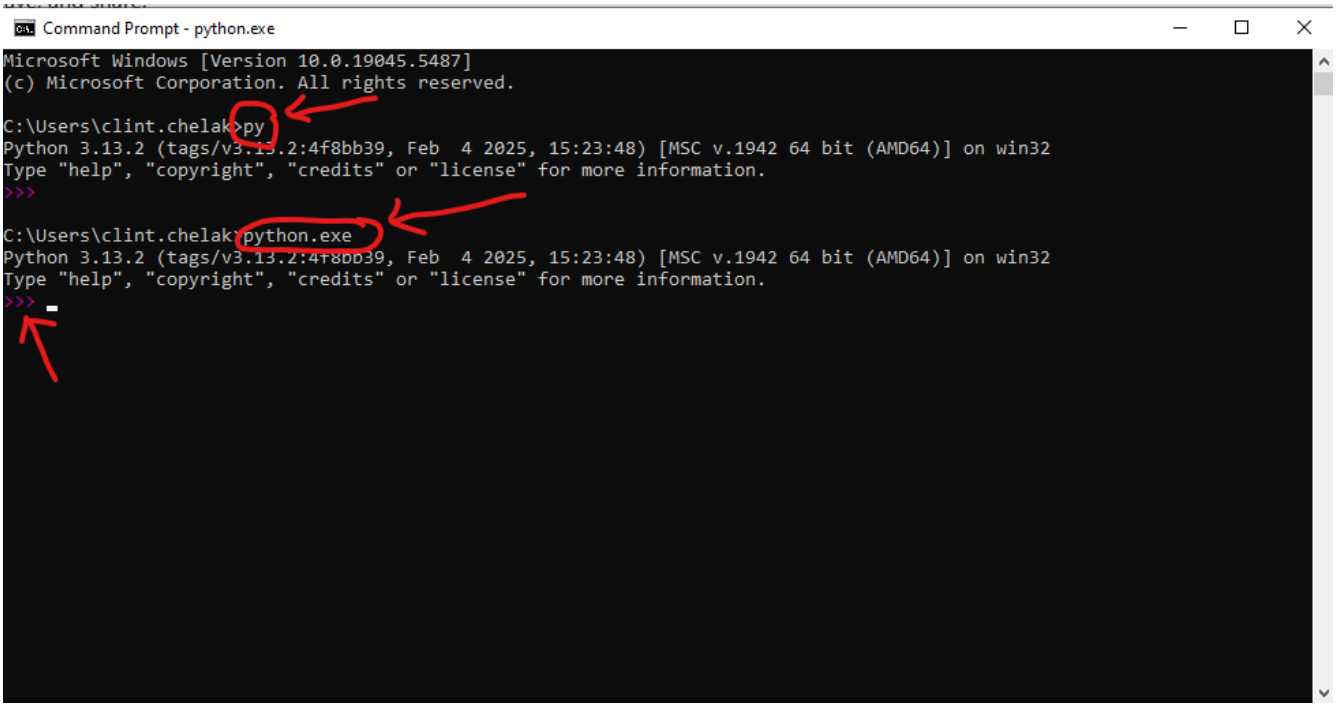




3. Ensure that python correctly installed by opening up the command prompt on your computer.



4. If during install, you checked the box to install “py launcher” type “py” into your console. Otherwise, type “python.exe”. If you see three brackets (“>>>”), you’ve arrived to the world of python!



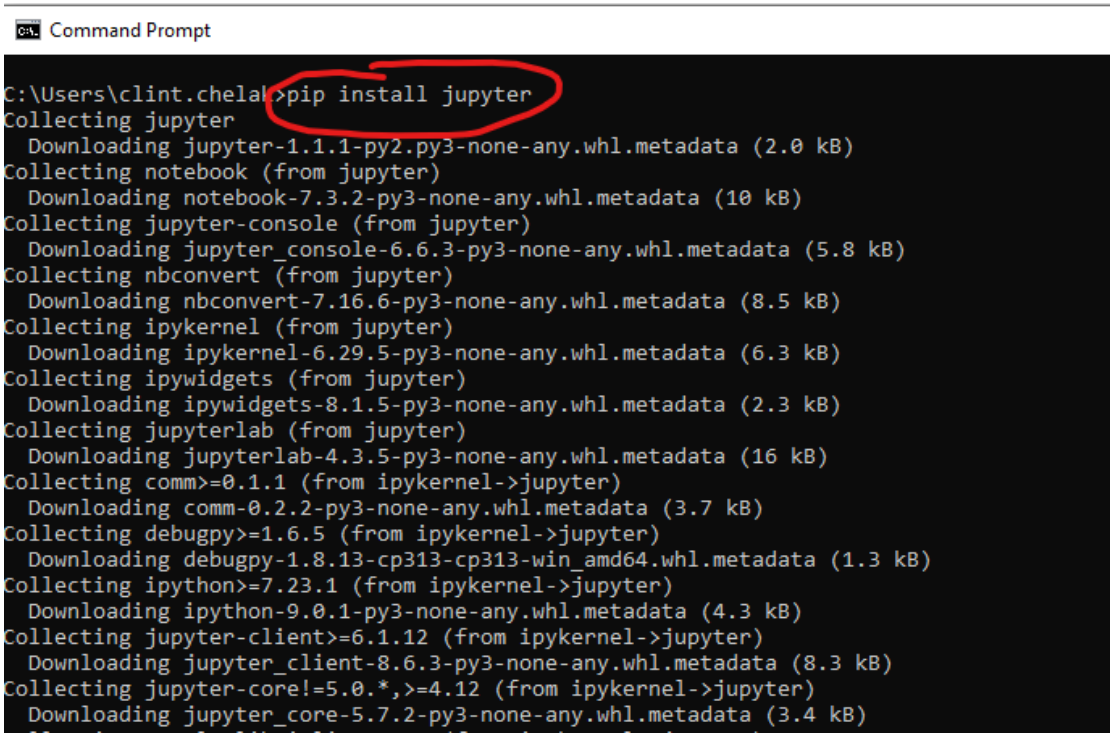
5. To exit the python console, type "`quit()`" or press "`Ctrl+d`". In this case, it will return you to normal command prompt functionality.

Opening a Jupyter Notebook

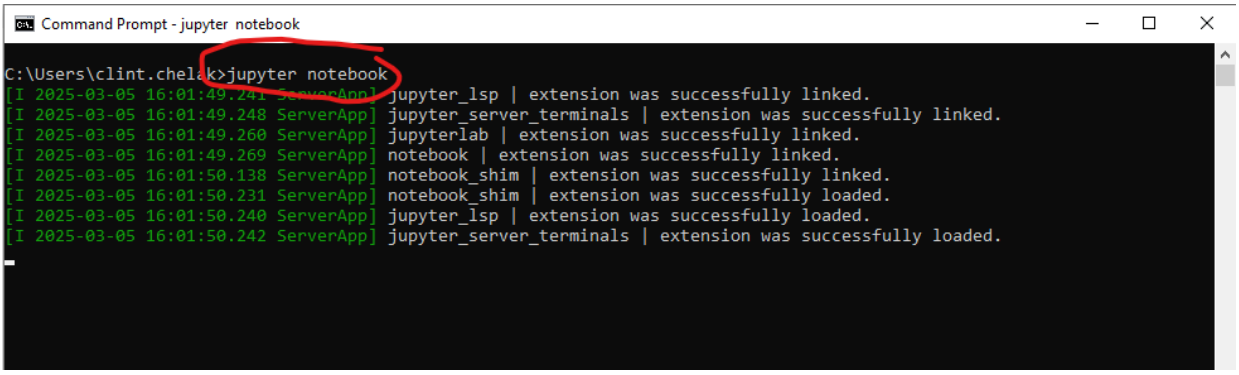
To work with jupyter notebooks, you need to install the jupyter package. Afterwards, you can use Jupyter notebooks from any IDE that supports them, such as Visual Studio Code. Or, as shown here, you can even open them in the same command prompt!

As with all python packages, we will install jupyter with pip.

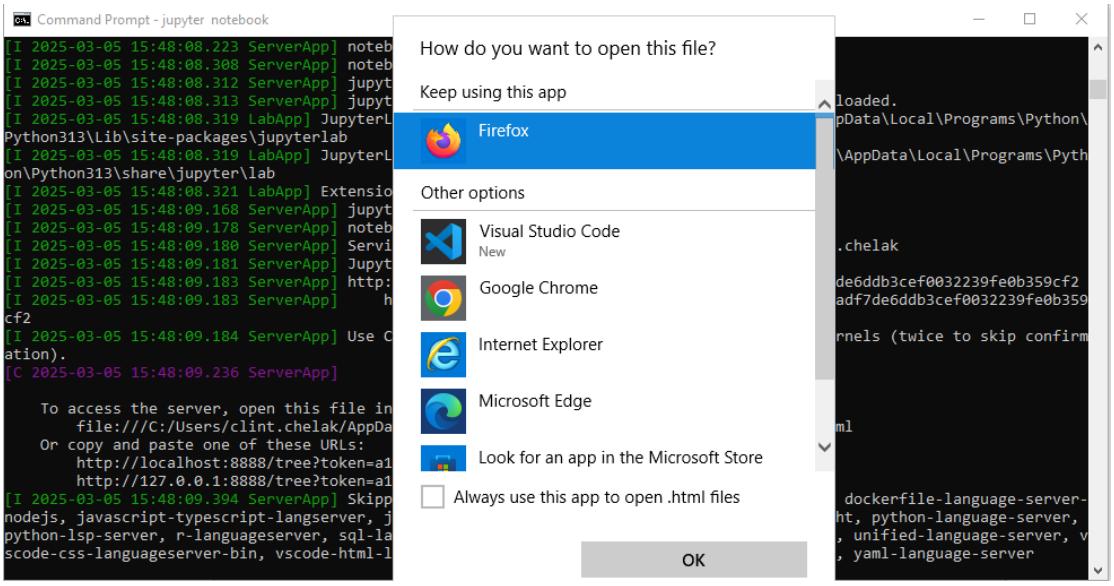
1. Open up the command prompt.
2. Type following command:
   `pip install jupyter`



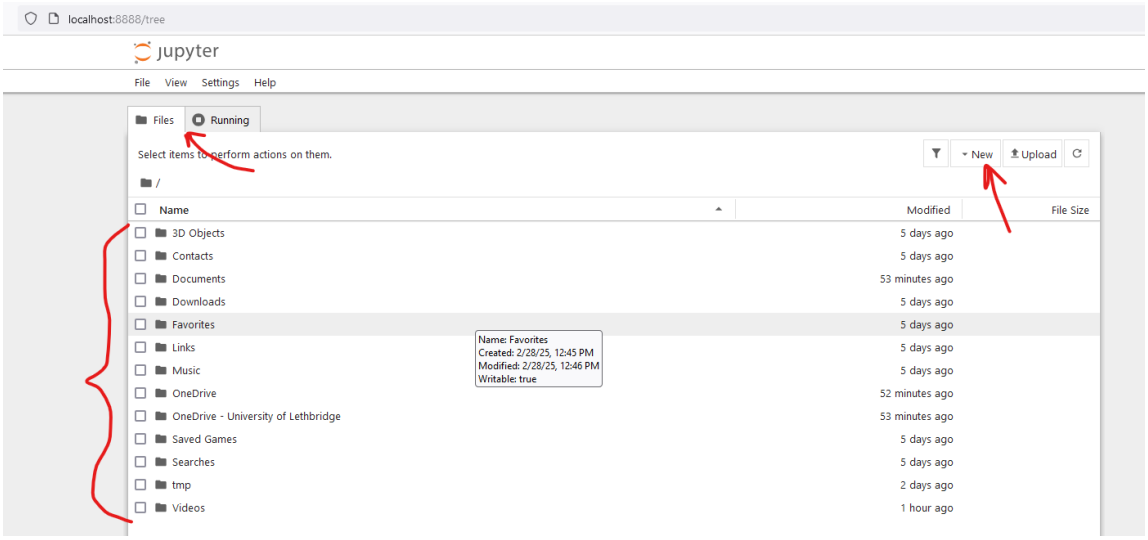
3. Wait for the command line prompt to reappear. Installing jupyter can take a while.
4. To confirm jupyter installed correctly, type the following into the command prompt:
   `jupyter notebook`



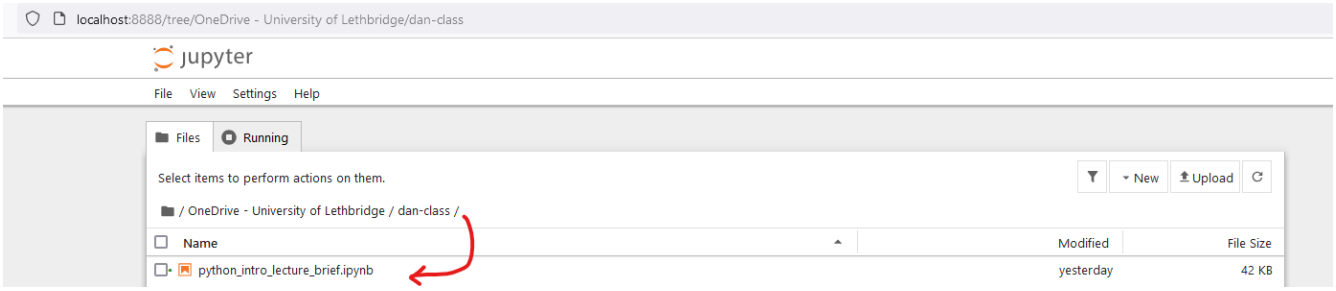
5. You may be asked which browser you want to open it with. Select the internet browser of your choice.



6. A menu appears showing your home directory and its files. If you have an existing notebook you want to open, navigate to where you saved the file. Otherwise, navigate to the folder of your choice and click “New” to create a new notebook.



7. Once you've navigated to the correct spot, click on the notebook (a file ending in ".ipynb"). You can now begin interracting with the notebook.

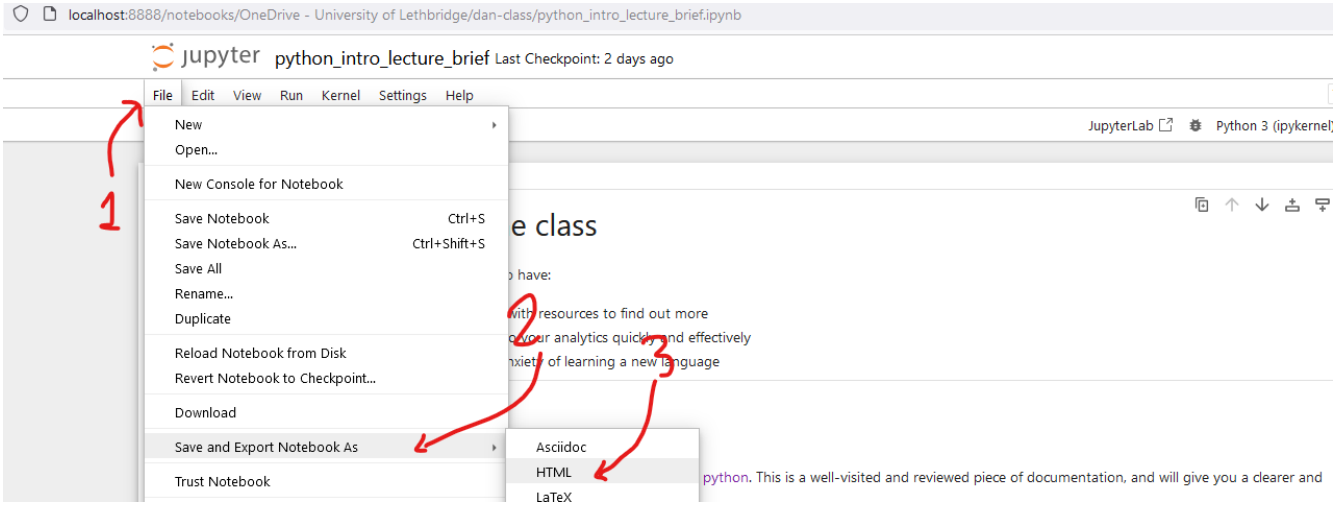


8. When finished editing your notebook, go up to File > Save, or press Ctrl + S.
9. You can close the browser, but you may notice the command prompt will still be running a Jupyter server. To close it,
    1. press Ctrl + C (or Esc), then type Y, then Enter
    2. or, press Ctrl + C twice, or press Esc twice
10. You can then close the command prompt.

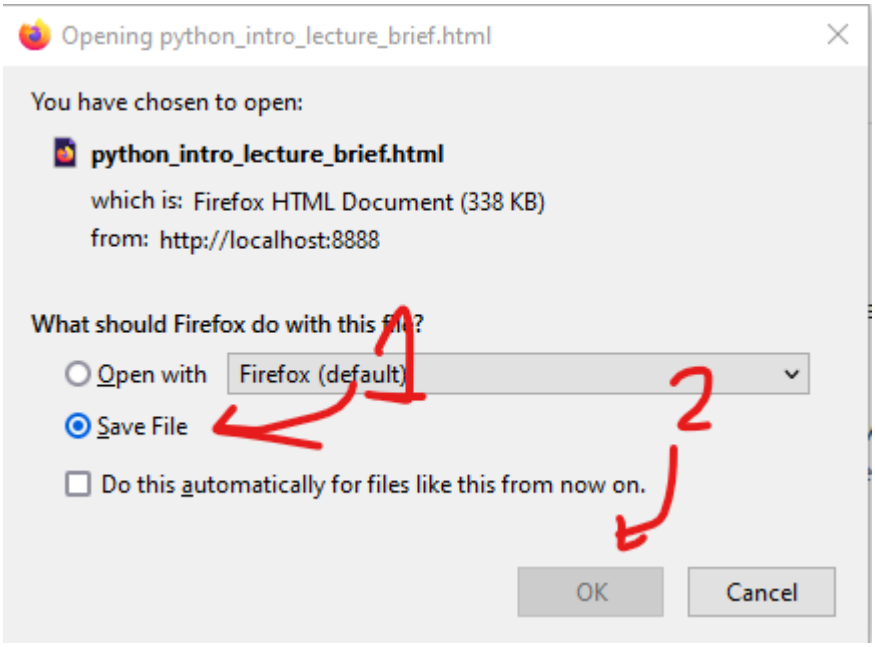
Exporting a Jupyter Notebook to HTML

Exporting your notebook to HTML is probably the easiest way to get it published. Without libraries, things like equations (found in the lab) will not render and will look like cryptic code, but at least the markdown text and code will look fine.

1. Open your jupyter notebook (see steps above)
2. Click File > Save and Export Notebook As > HTML



3. Depending on browser, it may download immediately, or give you a prompt like below.



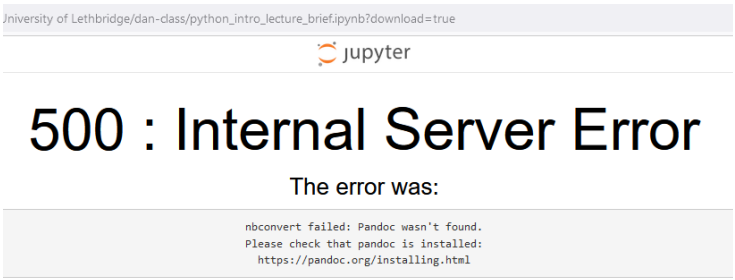
4. Open the HTML file. It should appear in a browser.
    1. Note: most internet browsers allow exporting an HTML file to PDF. Usually if you go to your browsers “Print” settings, you can convert it to the file type of your choice

Installing tools to Export Jupyter Notebooks to PDF

Depending on what is inside your notebook, you may need to install other tools before things will export to PDF properly. For example, on a Windows machine, the following will most likely need to be downloaded and installed:

- pandoc – a free, opensource document converter. Jupyter calls this to make PDFs.
- xelatex – a free python LaTeX library and exporter. Jupyter calls this to convert its markdown text format to PDF format.
    - LaTeX is a free (and arguably more powerful) alternative to Microsoft Word, designed for making professional documents. It does have a steep learning curve if you’re using it for something other than Jupyter, as it has its own coding and formatting syntax.

Jupyter will give an error message like the following whenever it needs a tool to continue.



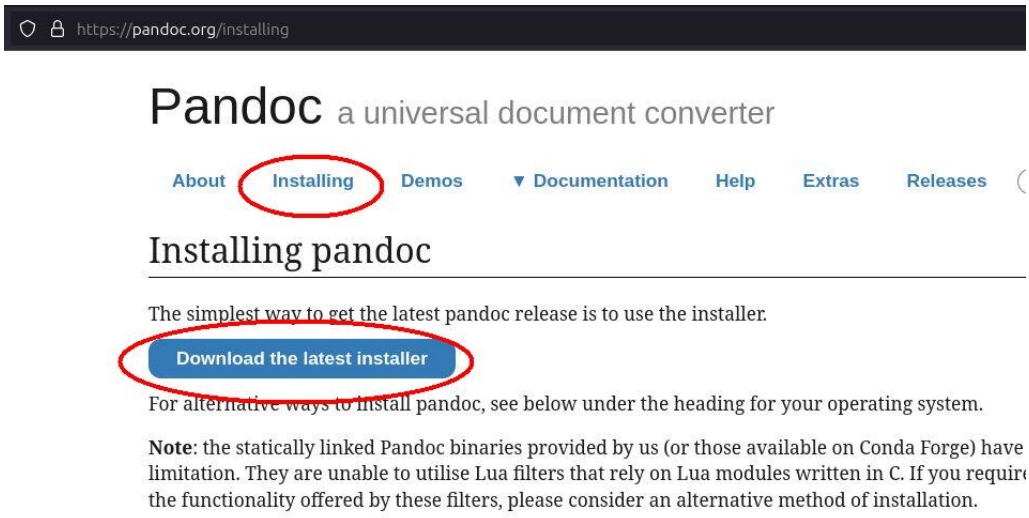
```
    raise OSError(msg)
OSError: xelatex not found on PATH, if you have not installed xelatex you may need to do so. Find further instructions a
t https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex.

C:\Users\clint.chelak\OneDrive - University of Lethbridge\dan-class>
```

Fortunately, the prompts give you a webpage to direct you to solution. I'd recommend following that. However, additionally, I have a guide below with images.

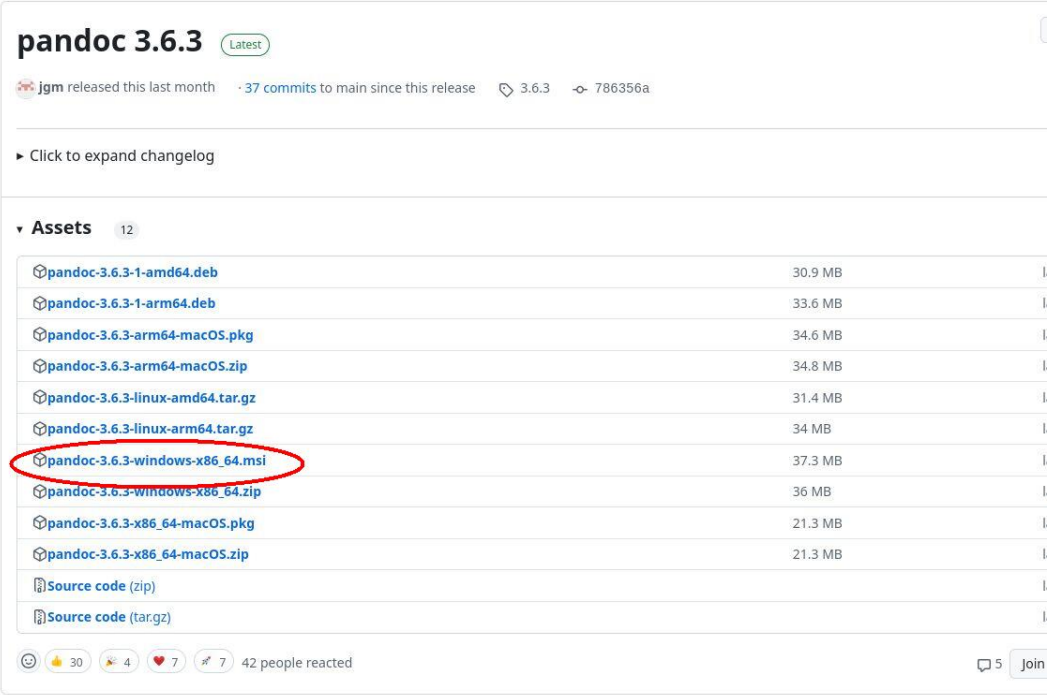
After installing any package, you will most likely need to restart your jupyter session and command prompt.

Pandoc

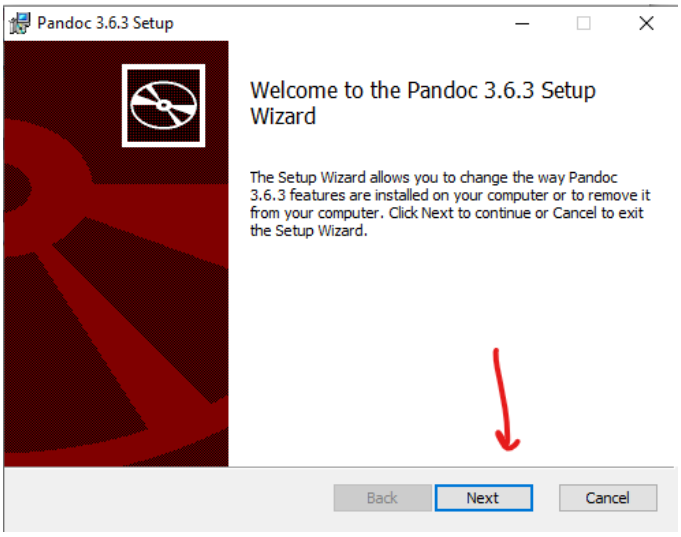
1. Navigate to the pandoc website
2. Click on their "Installing" tab
3. Scroll to the windows section, click the link to the download page. It will probably bring you to a github page like the one linked here.



4. For Windows, download the file ending in "-x86_64.msi". The windows installer will be downloaded.



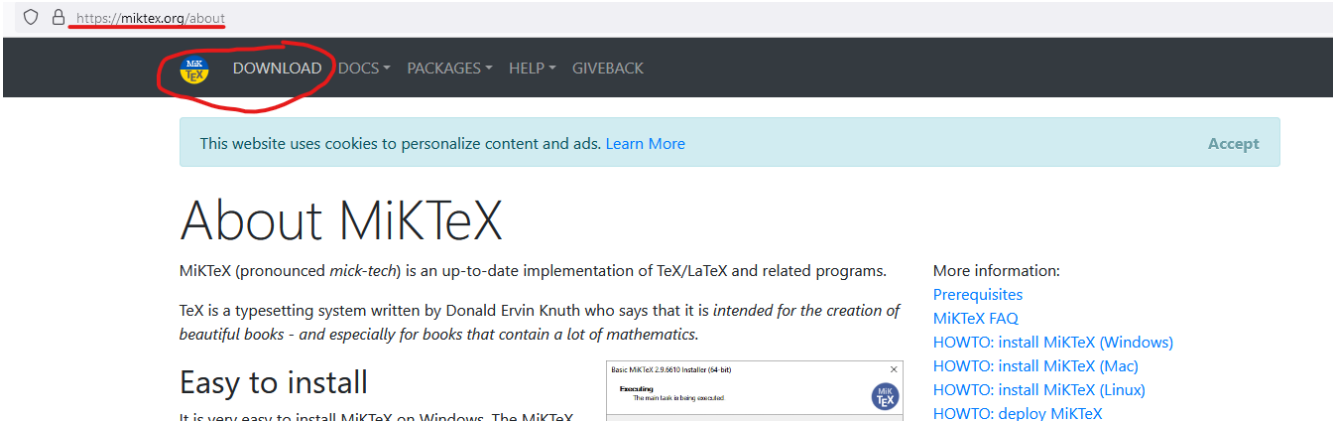
5. Follow the steps of the installer, similar to what is seen below



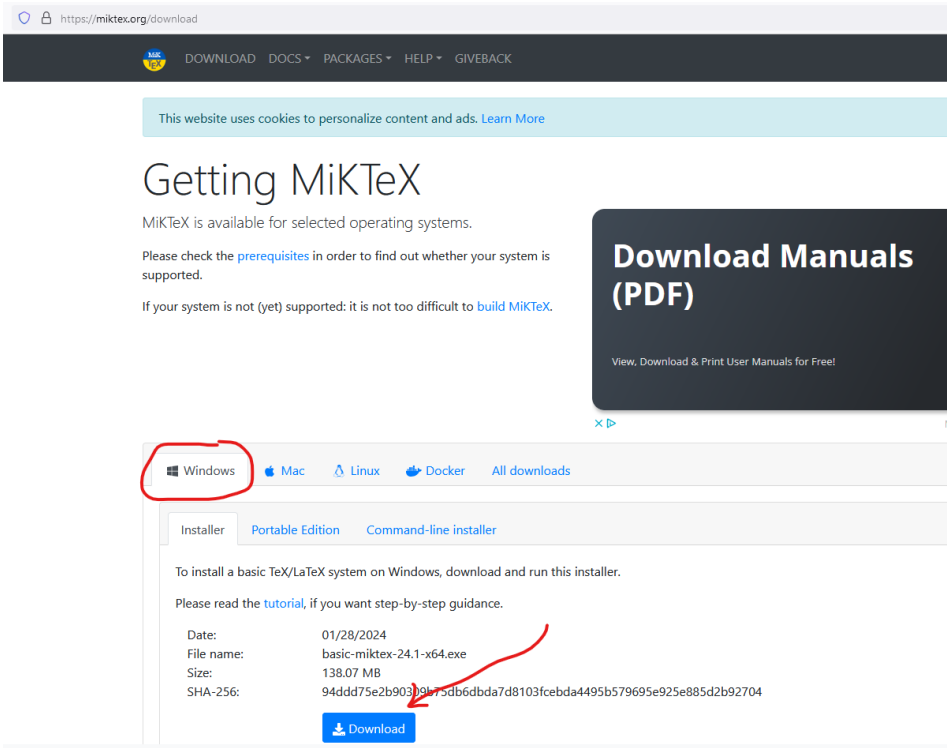
Miktex

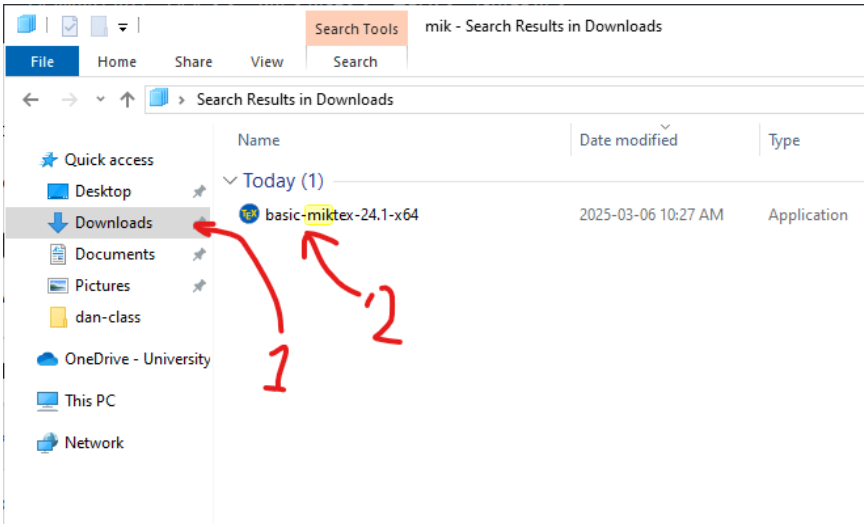
Miktex is one of the LaTeX tools with Windows support. I wasn't a fan of this one or the other one I tried for Windows, but I guess this is what we've got.

1. Go to miktex home page. Click the "Download" tab

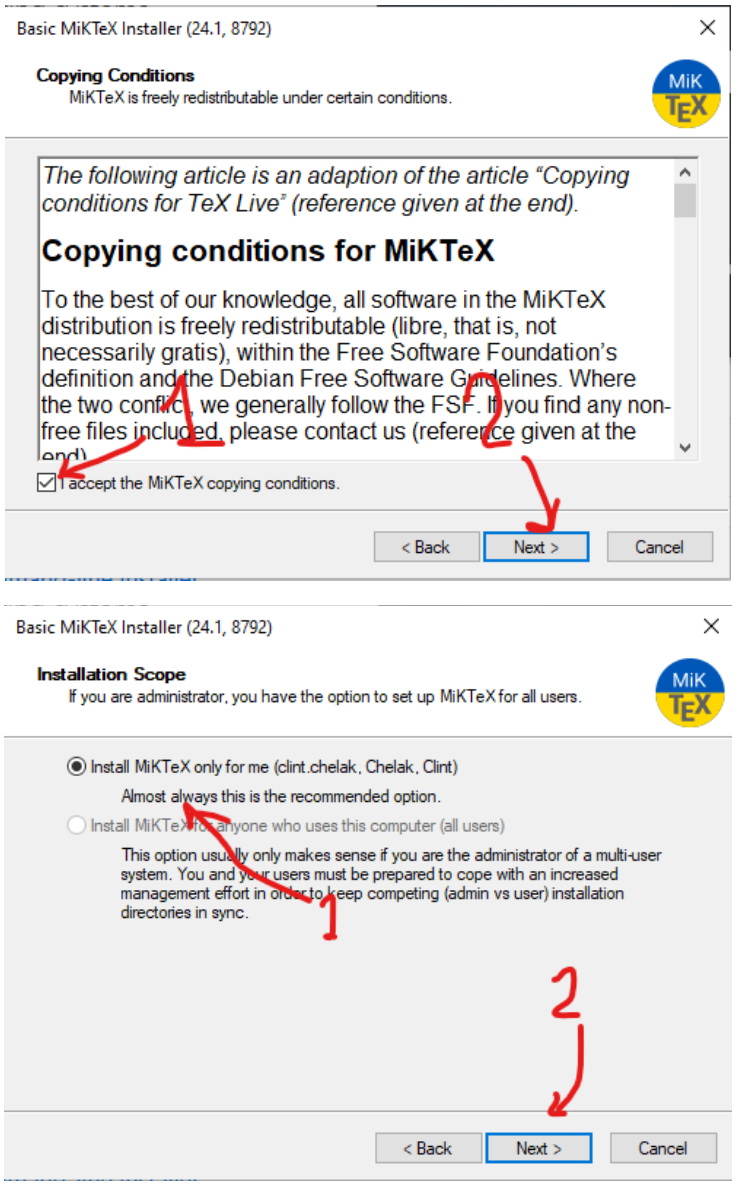


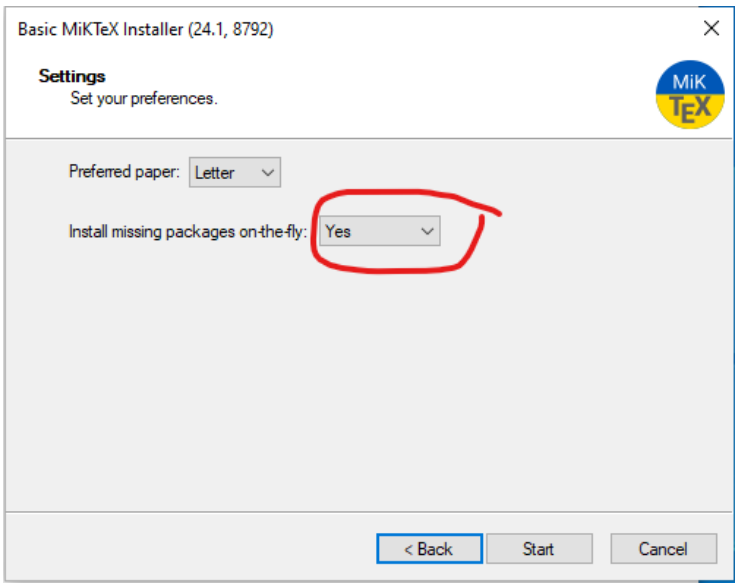
2. Click the “Windows” tab, then click “Download”. Open the installer from your Download folder.

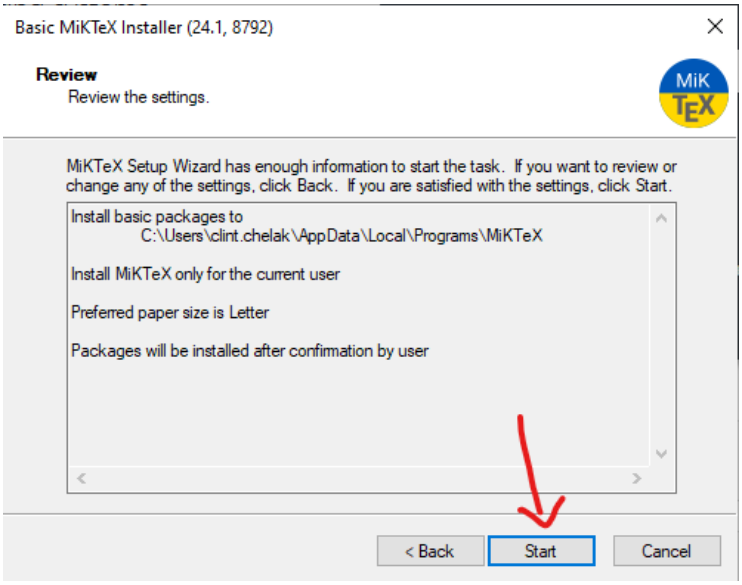


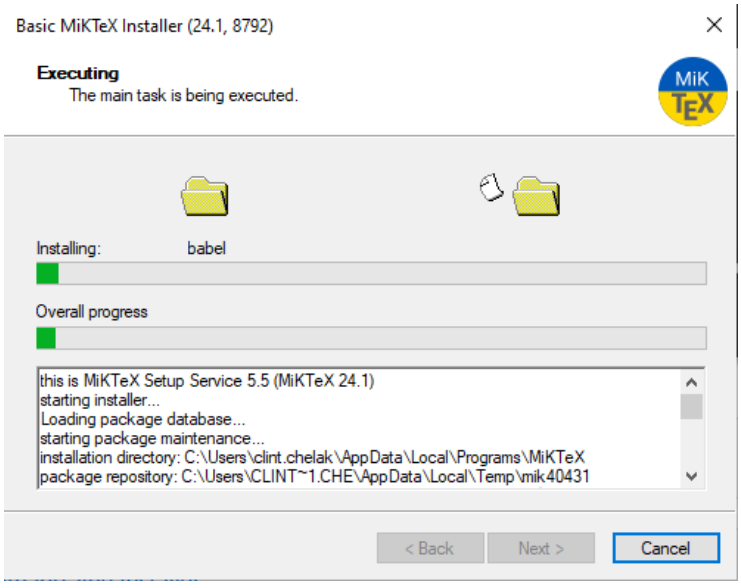


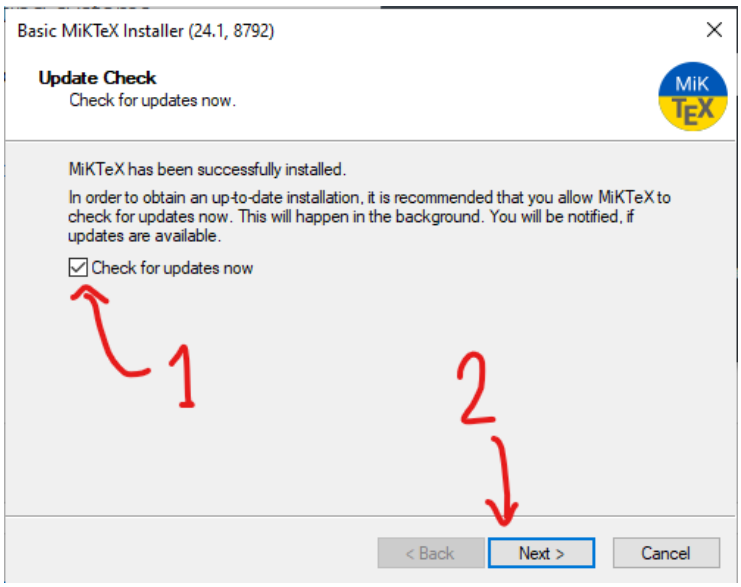
3. Follow install steps similar to below

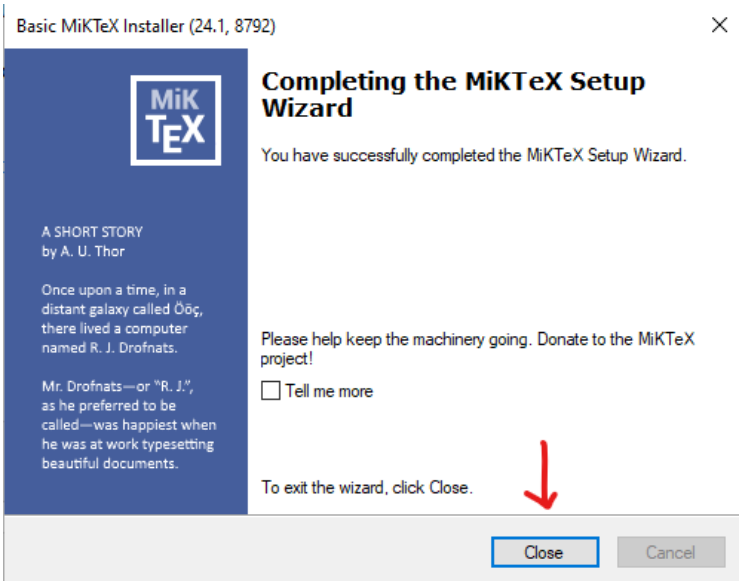



Basic MiKTeX Installer (24.1, 8792)
Settings
Set your preferences.
Preferred paper: Letter
Install missing packages on-the-fly: Yes
< Back
Start
Cancel

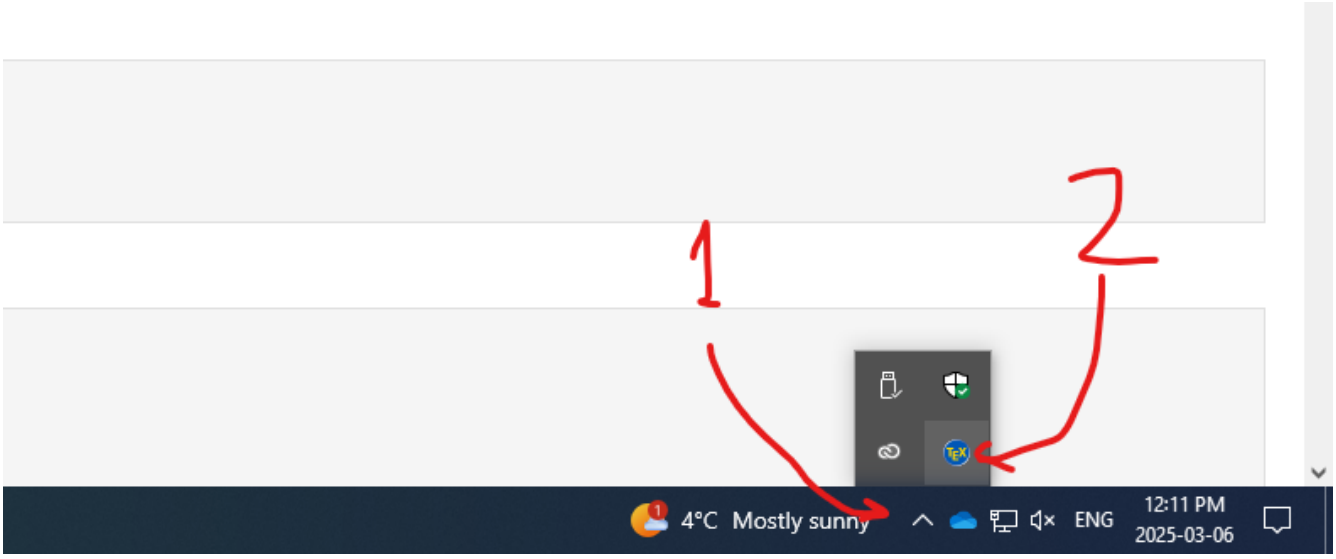

Basic MiKTeX Installer (24.1, 8792)
Review
Review the settings.
MiKTeX Setup Wizard has enough information to start the task. If you want to review or change any of the settings, click Back. If you are satisfied with the settings, click Start.
Install basic packages to
C:\Users\clint.chelak\AppData\Local\Programs\MiKTeX
Install MiKTeX only for the current user
Preferred paper size is Letter
Packages will be installed after confirmation by user
< Back
Start
Cancel

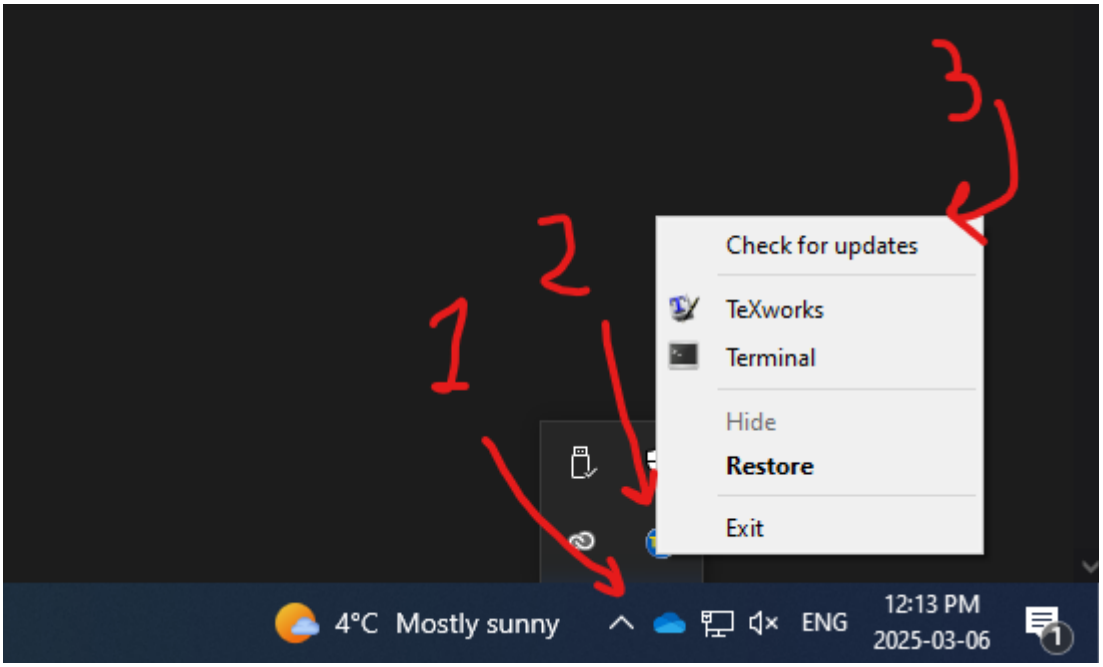

Basic MiKTeX Installer (24.1, 8792)
Executing
The main task is being executed.
Installing: babel
Overall progress
this is MiKTeX Setup Service 5.5 (MiKTeX 24.1)
starting installer...
Loading package database...
starting package maintenance...
installation directory: C:\Users\clint.chelak\AppData\Local\Programs\MiKTeX
package repository: C:\Users\CLINT~1.CHE\AppData\Local\Temp\mik40431
< Back
Next >
Cancel



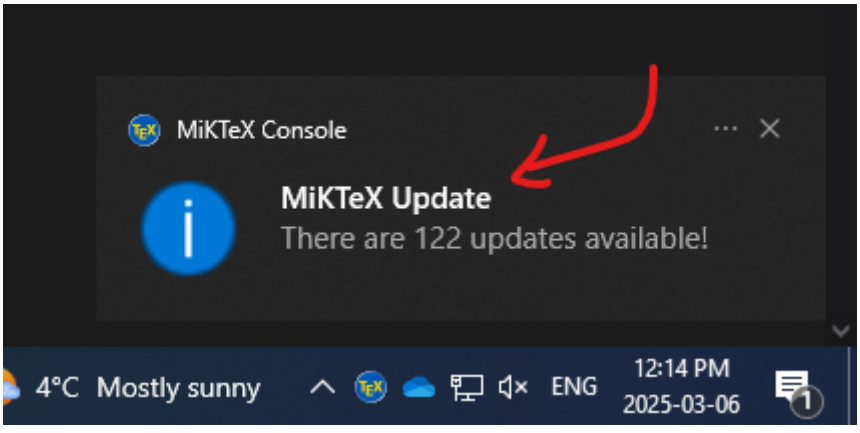


4. If you didn't check "Check for updates now" in the installer, or didn't see a popup like below that updates are avaliable, run the updater by clicking the drop menu on the bottom of your windows tab and right-clicking the Miktex icon.

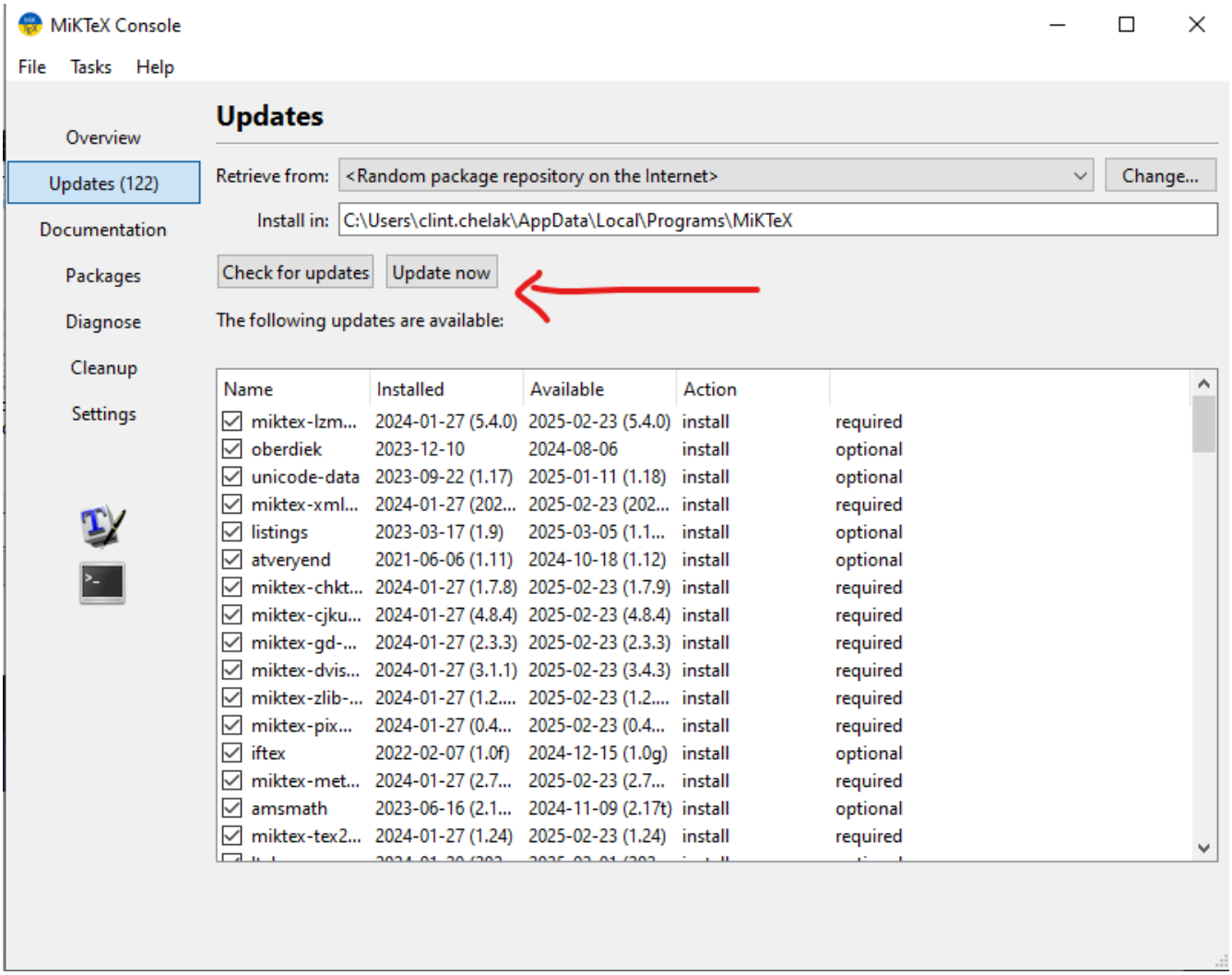




5. You should see a popup saying a “MiKTeX Update” is available. Click on it. It should start up the Miktex console.



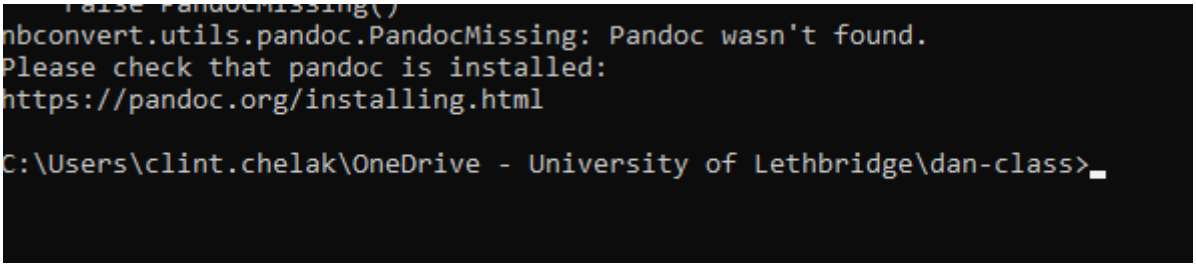
6. When open, click Update



7. After the update, Miktex will inform you that it needs to close. Close down Miktex
8. After installing this and pandoc, restart Jupyter by stopping any processes in your command line or in your IDE (or whatever else you're using as a tool).

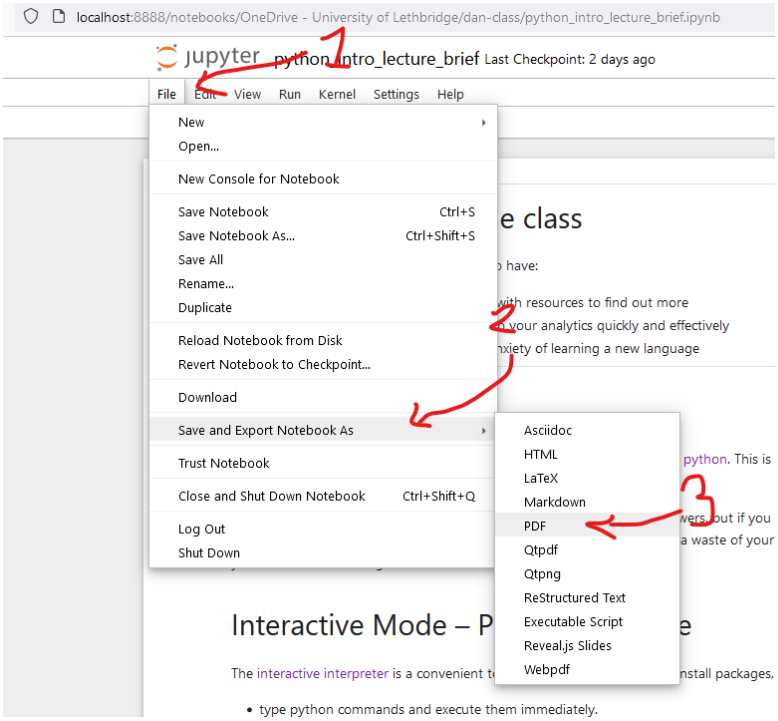
Exporting a Jupyter Notebook to PDF

There are three good options for exporting a jupyter notebook to a pdf or HTML file from what we learned so far. All are listed below. If at any time you are getting errors like the ones seen below, visit the "Installing tools..." section above.

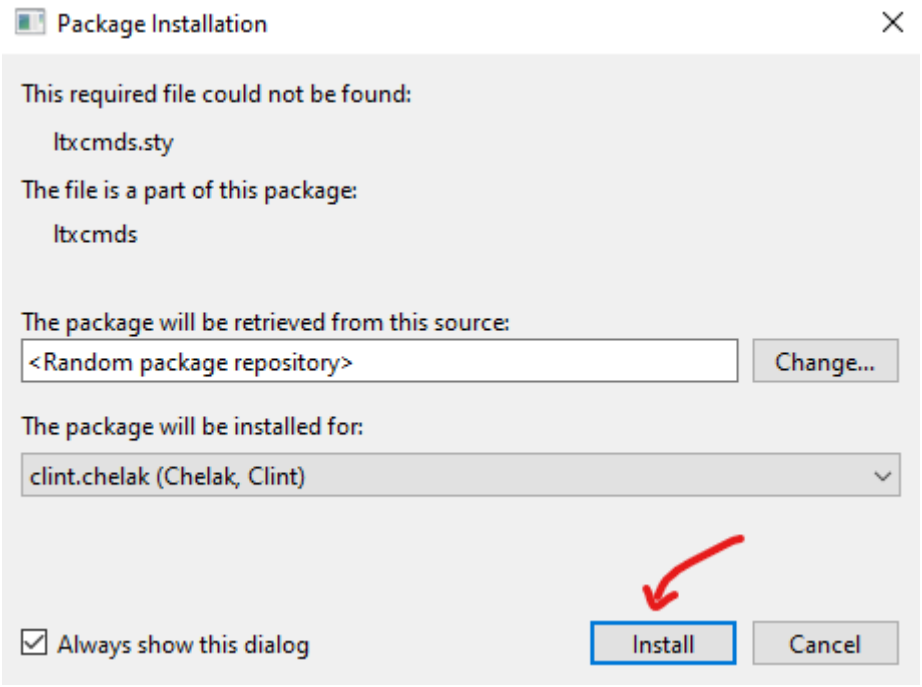


Through Jupyter Notebook IDE

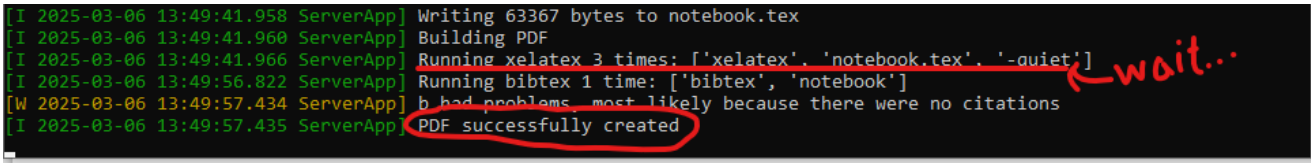
1. Open the jupyter notebook you wish to export in the browser (see steps in section above)
2. Click File > Save and Export Notebook As > PDF



3. A new tab will open. It will be loading for some time. Under the hood, it should be performing the export. **If it hangs, check for a window like the following.** The LaTeX tool may need to install some more libraries.



1. On your first run, a popup may appear many times. Feel free to click that check box saying not to show it every time. It will install the packages automatically in the background. It did leave me sitting on my hands for a while, though.
2. The first export took a long time for me. If it takes longer than 5 minutes, return the command line and press the arrow key to see if it updates.
3. When finished, the logs on the command line will print that it successfully exported



4. By default, the PDF should be saved to Downloads. Check your downloads file for a PDF matching the name of the Jupyter notebook (i.e. the "`.ipynb`" file).

Through the Windows Command Prompt

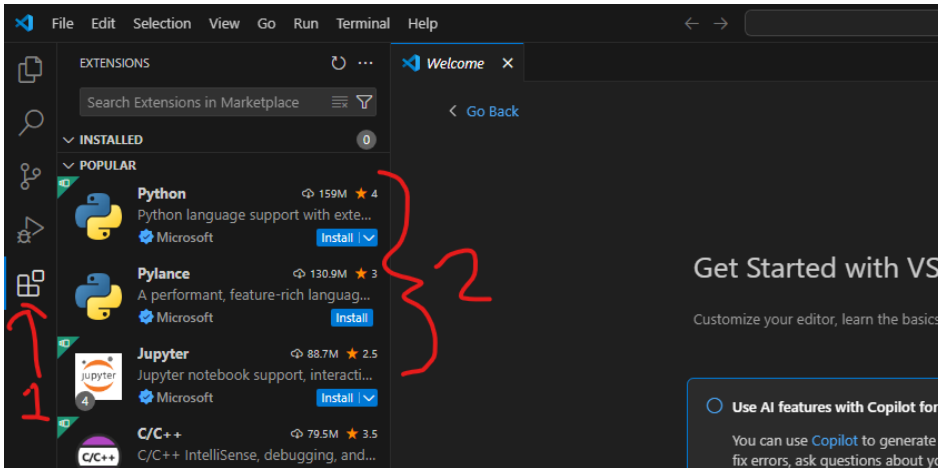
Exporting your file through command prompt can be a single line, and is probably the simplest, provided you know how to navigate the command prompt.

1. Open the command prompt (see end of python installation steps)
2. Type the following into your browser, substituting the example path with your notebook path:
   `jupyter nbconvert --to pdf C:\path\to-my\project\my_lab.ipynb`
   1. Note that it will save into the same folder as your notebook.

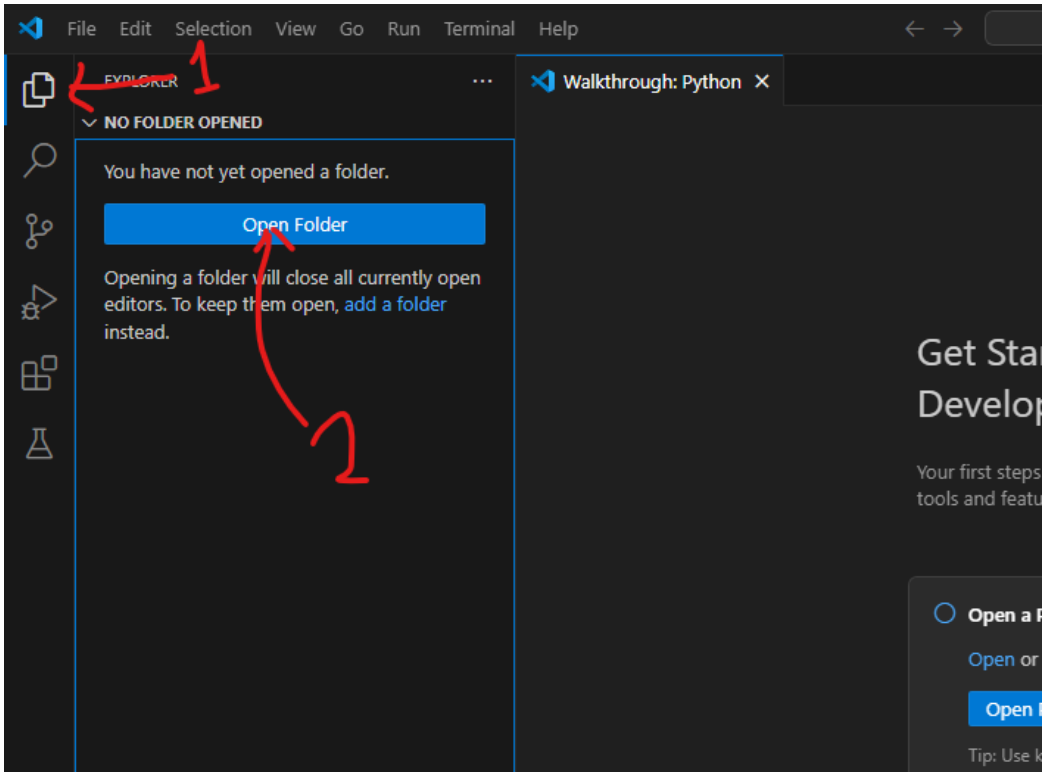
Through Visual Studio Code

With the Jupyter extension installed, exporting is built in on the file display.

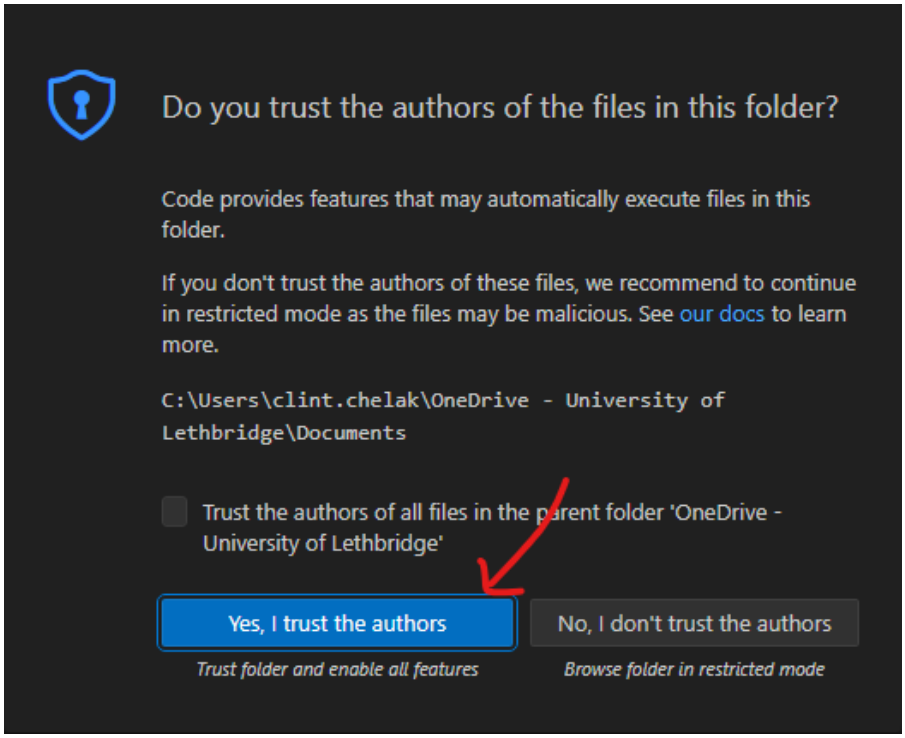
1. Open up the Visual Studio Code application from the start menu
2. Go to extensions. Install the Python and Jupyter extensions



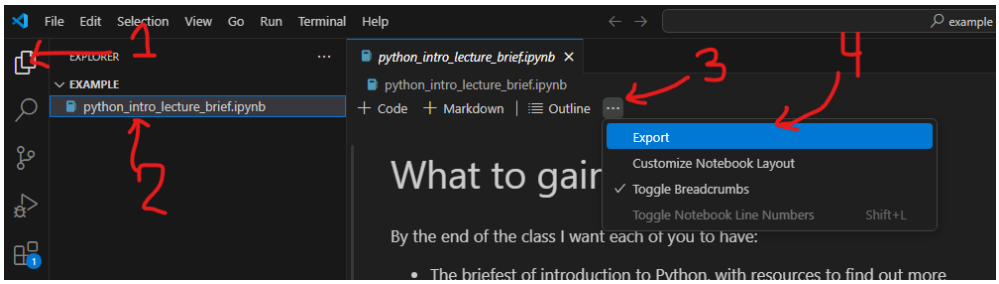
3. Click the folder icon. Click open folder. Navigate to the folder where you saved the "`.ipynb`" file.



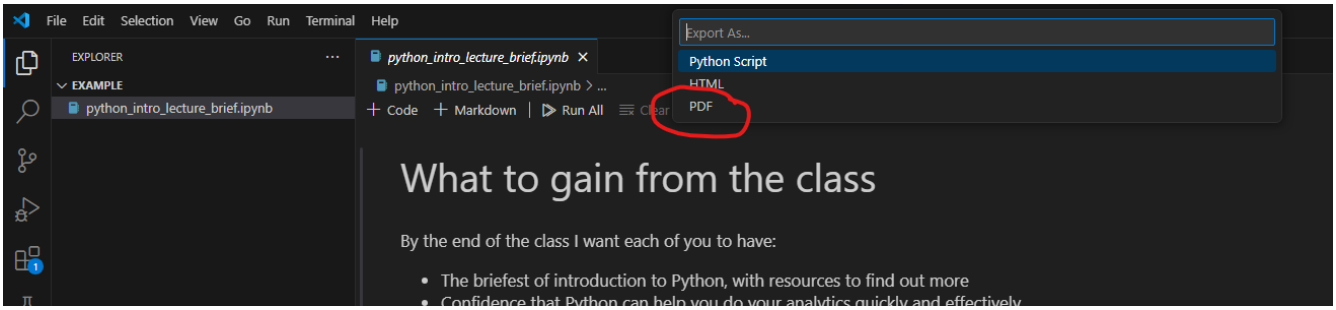
4. When it opens, a menu will ask if you trust the authors to of that folder, click that you do.



5. You should now see your folder displayed on the left tab, with your ".ipynb" file there. Click on it. Your file should open. At the top of the file, you should see an icon 3 dots ("..."). Click it, then click export.

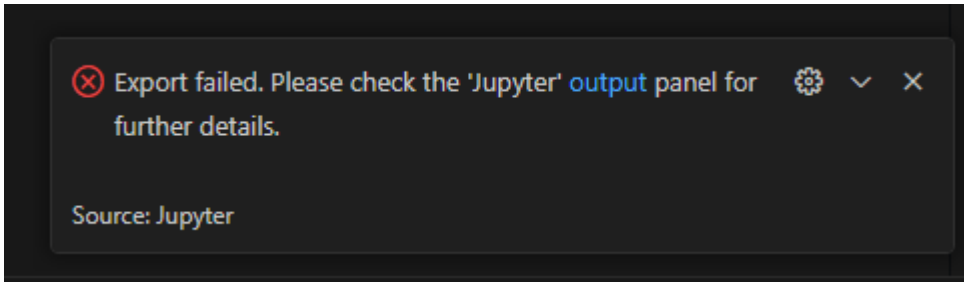


6. A drop-down menu from the top should appear. Click the export format you want (PDF).

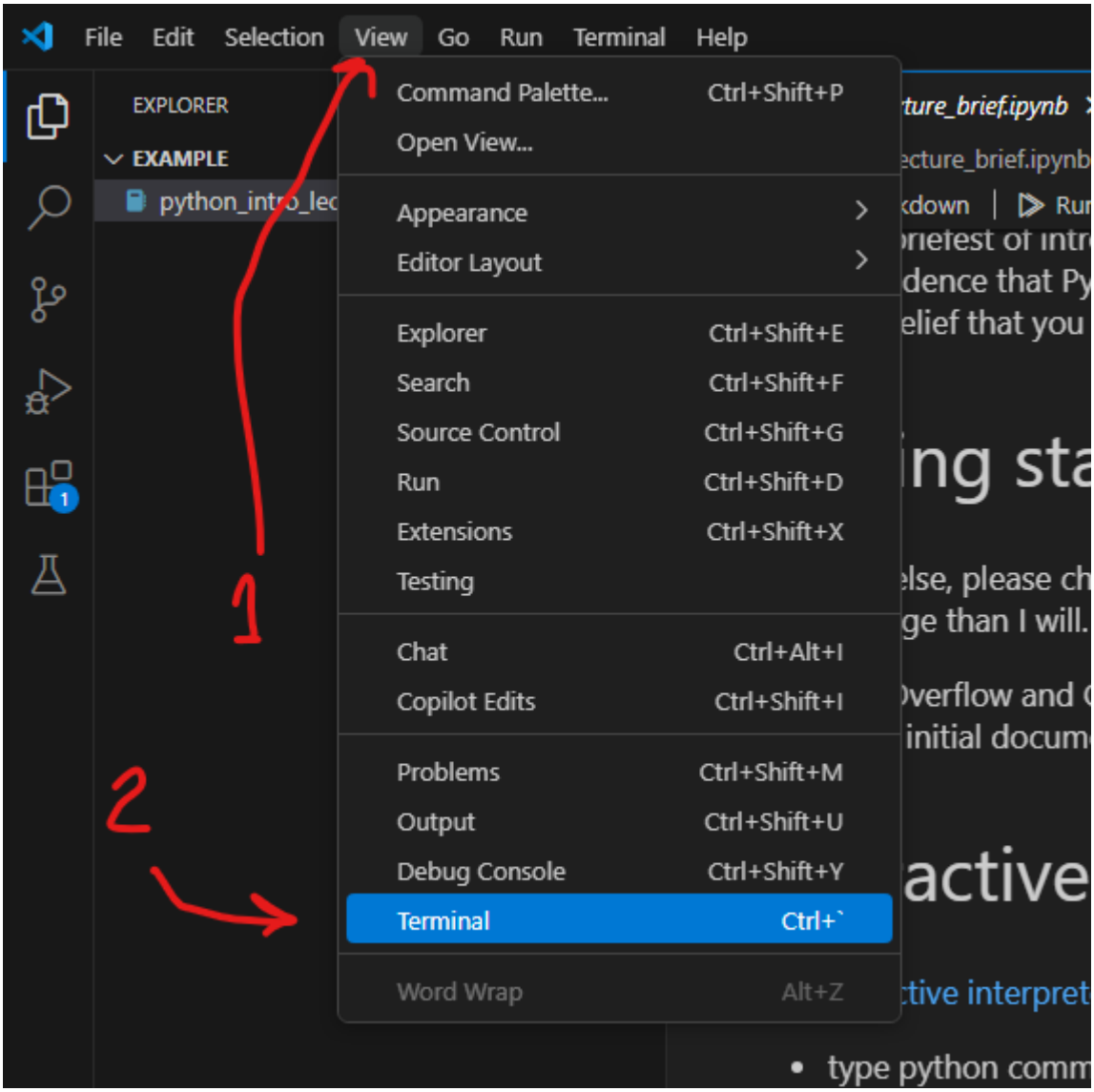


In the event of a failure during export, you will see a pop-up on the bottom-right corner of Visual Studio code. It gives instructions on how to see errors. I'll give an example below.

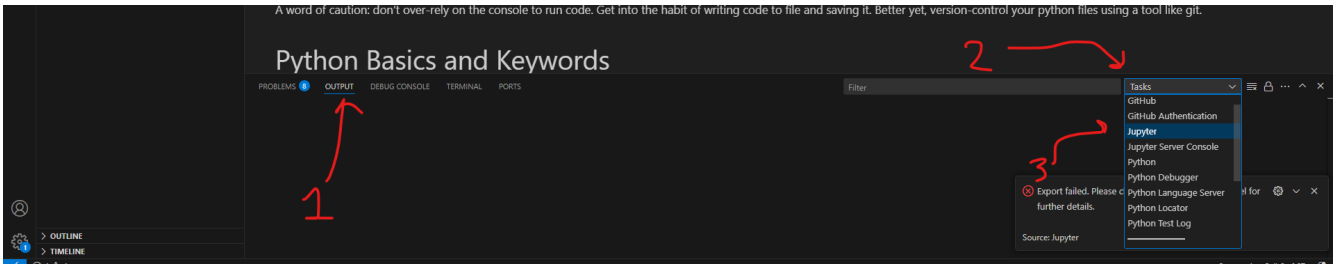
1. If a message like the following appears, we'll need to see the log output.



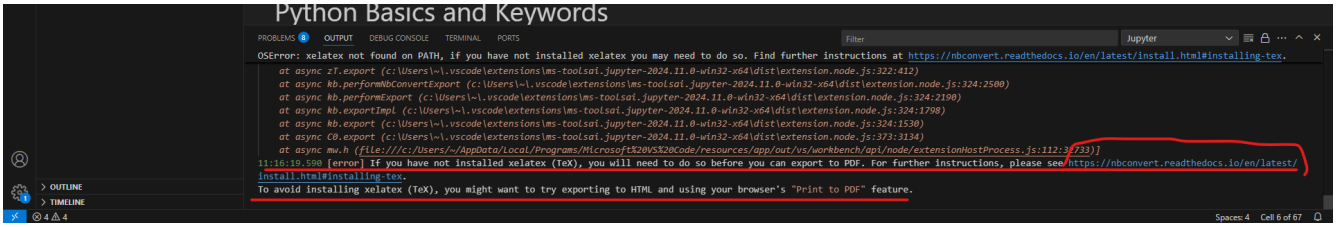
2. If you don't see a "Terminal" menu at the bottom of your screen, we should open it. Press `Ctrl + ` or click View > Terminal



3. On the bottom of your screen in the terminal menu, navigate to the Jupyter tab as follows:
    1. Click the “Output” tab
    2. Click the drop-down on the right, probably saying “Tasks”
    3. Navigate to the “Jupyter” option in the drop-down and click it.



4. Logs should appear in the output menu now, as seen below. Read the bottom lines to get a summary of what went wrong. For example, here I’m missing some pdf libraries.



As with just about everything with Visual Studio Code, there are shortcuts to get to these menus quicker. Review Visual Studio Code documentation to see results in a few key presses.