Using TeX in Python – Troubleshooting Guide Updated June 6, 2023

Sinclair Combs

First, I **highly** recommend installing MiKTeX on your computer (https://miktex.org/). This is an open-source, integrated TeX package manager that automatically flags missing required packages and allows you to view locally-installed packages.

If you're getting an error when trying to implement TeX into any Python code (i.e., matplotlib notebooks), here are a few things to try:

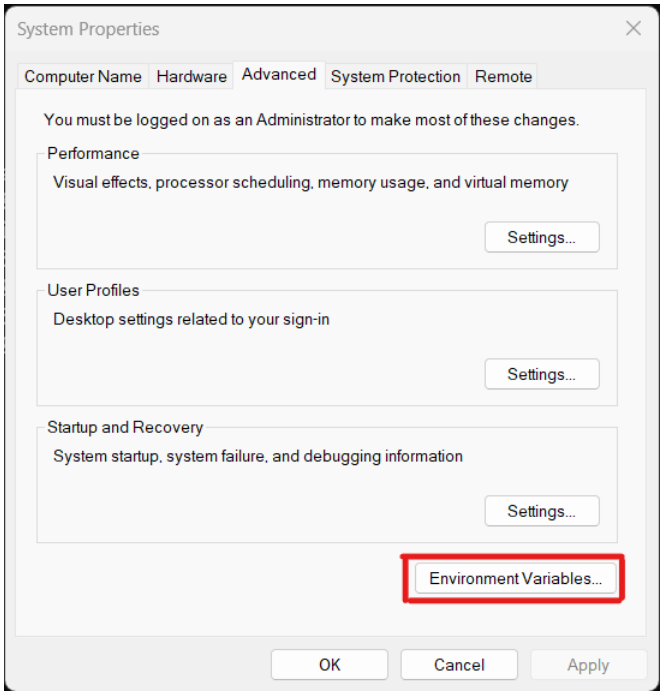
1. Check if TeX / LaTeX is installed

   → Enter `pdftex --version` or `pdflatex --version` in the command line

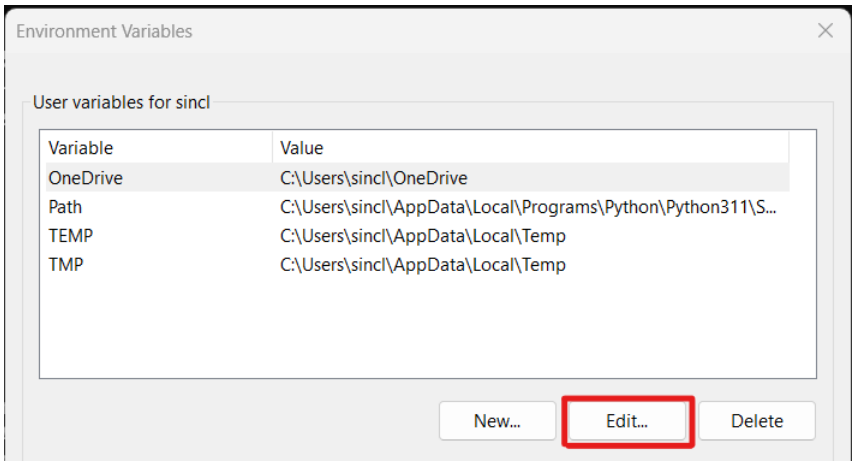
2. Check if Python recognizes external dependencies

   Instructions for Windows

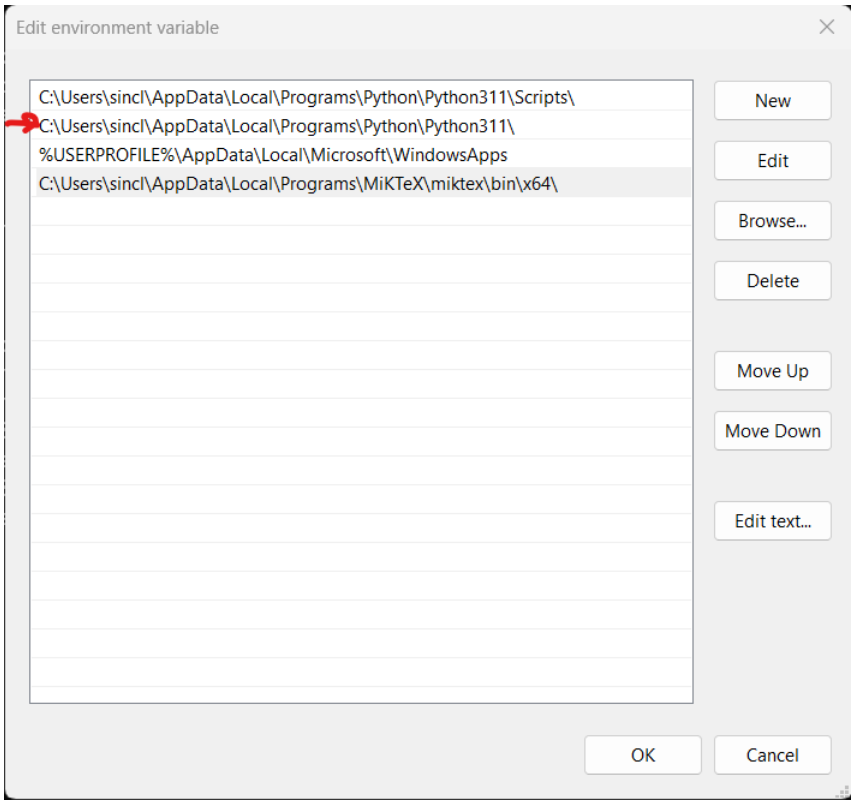
   → Open Advanced System Properties to view Environment Variables, this can be done by entering "Edit the system environment variables" into the Windows search bar. Select Environment Variables.



→ Select Edit under the user variables section



→ See if there is an environment variable with a path to the Python directory. If not, add a new environment variable with that path. Likely, it will be something similar to
`C:\Users\sincl\AppData\Local\Programs\Python\Python311\`



Instructions for macOS / Linux

→ To list the current list of Python directories: `echo $PYTHONPATH`

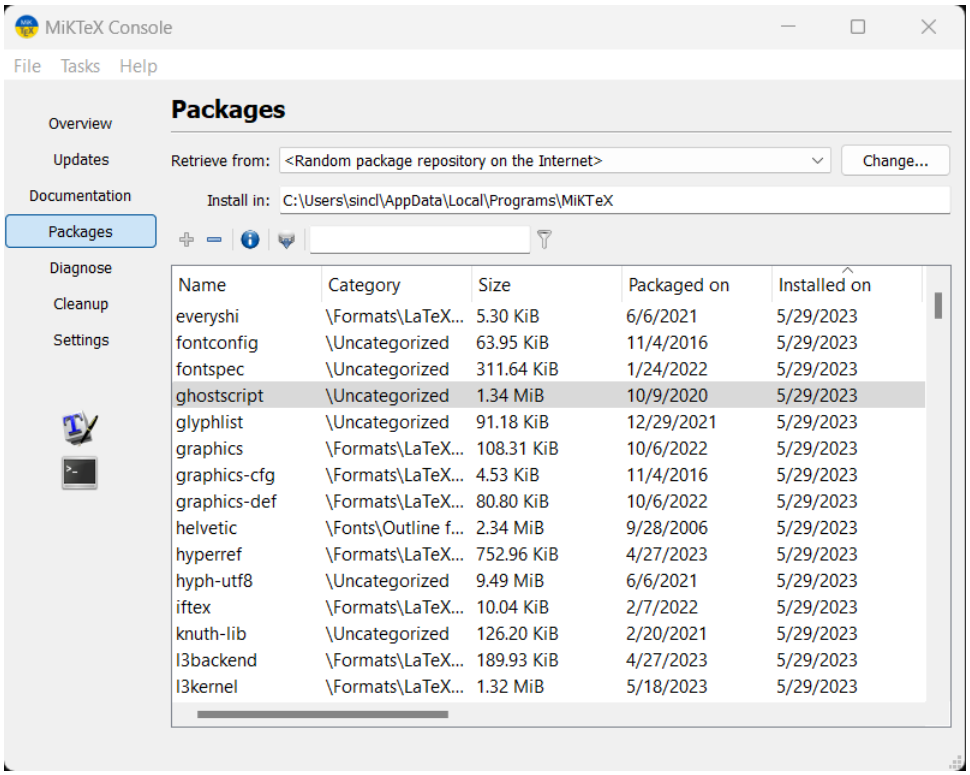
→ Setting environment variables depends on what your default shell is. Common shells include `bash` and `csh`. To check default shell: `echo $SHELL`

→ To create a new environment variable:
`export PYTHONPATH=~/Python` (`bash/ksh` shell)
`setenv PYTHONPATH ~/Python` (`csh/tcsh` shell)

→ To prepend to an existing environment variable:
`export PATH=~/bin:${PATH}` (`bash/ksh` shell)
`setenv PATH=~/bin:${PATH}` (`csh/tcsh` shell)

3. Check if `ghostscript` and `dvipng` (required TeX packages) are installed

→ This can be done through the MiKTeX console



Other useful information:

→ Make sure you're using math mode (indicated by $$ environment) when appropriate

→ All text requires valid LaTeX syntax or an error will be thrown