

WRF-Python & Jupyter Notebooks

A collection of diagnostic and interpolation routines for use with output from the Weather Research and Forecasting (WRF-ARW) Model.

<https://wrf-python.readthedocs.io/>

Configuration on Windows 10 for NVU-Lyndon Atmospheric Sciences

Miniconda (a Python package manager) is already installed in ASAC 315 and the WX Center. Using the conda-forge channel and installing additional Python packages is necessary for WRF-Python to work. This documentation also includes instructions on starting the Jupyter notebook user interface.

1. Click on the Start menu and search for **Anaconda**. Click on the **Anaconda Prompt**
2. Type or copy the command below into the Anaconda Prompt. You should run this command even if you have already done it in the past. This will ensure that conda-forge is set as the highest priority channel. The double hyphens before “add” are critical.

```
conda config --add channels conda-forge
```

About conda-forge

Conda-forge is a community-driven collection of packages that are continually tested to ensure compatibility. WRF-Python developers highly recommend using conda-forge when working with conda. See <https://conda-forge.github.io/> for more details.

3. Install the necessary Python packages by typing into the Anaconda Prompt:

```
conda install jupyter matplotlib cartopy netCDF4 wrf-python
```

This command will automatically determine if any additional package dependencies are required, and download and install them as well.

4. When prompted, type: **y** and press Enter to proceed. It may take a couple minutes to install everything.
5. Change directories to where you have saved the wrfout netCDF files. For example, type:

```
cd C:\Users\\Desktop
```

It is easiest to work with wrfout netCDF files when running the WRF-Python code (and Jupyter notebooks) from the directory where the wrfout files are saved. The Anaconda Prompt on Windows acts very similar to a Terminal in Linux, where it recognizes the cd command to change directories.

Windows vs. Linux

Windows uses backslashes: \ between directories. Linux uses forward slashes: / Additionally, Windows is not case-sensitive, Linux is always case-sensitive.

6. To launch Jupyter, type the following into the Anaconda Prompt:

```
jupyter notebook
```

This will launch a new browser window (or a new tab) showing the Notebook Dashboard, a

sort of control panel that allows (among other things) to select which notebook to open.

By default, Jupyter can access only files within the folder it was started from (including any sub-folders). This is why step 5 changed directories to the directory where the wrfout files are saved (this simplifies reading the netCDF files within the Python code).

Modifications to the notebooks are automatically saved every few minutes. When opening a pre-written Jupyter Notebook, to avoid modifying the original notebook, it is recommended make a copy of the notebook document first (menu File → Make a copy ...) and save the modifications on the copied notebook.

Warning

Pay attention to not open the **same** notebook document on **many tabs**: edits on different tabs can overwrite each other! To be safe, make sure you open each notebook document in only one tab. If you accidentally open a notebook twice in two different tabs, just close one of the tabs.

How to Close Jupyter Notebooks

Closing the browser (or the tab) **will not close** the [Jupyter Notebook App](#). To completely shut it down you need to **close the associated terminal** (the Anaconda Prompt).