

2023 Sagan Summer Workshop

Characterizing Exoplanet Atmospheres: The Next Twenty Years

Getting Started Guide: Python

For the workshop hands-on sessions, participants can choose to run the exercises from their own installation of Python or use Google Colaboratory. The instructions in this document are for using your own installation of Python.

Please note that *you should set up your Python distribution in advance of the workshop*; just follow the instructions below. In this document, we use the `Courier New` font to indicate a command.

Each hands-on session will have its own notebook, which includes the setup notebook and instructions for file downloads needed for that session. Please read through the whole document first before you start installation.

The Python Distribution

To run the notebooks as Jupyter notebooks used in the hands-on session, you first need a working installation of Python3. Instead of the latest version of Python (3.11.X) we recommend version **3.9.X or 3.9.7** for the hands-on sessions. Even if your operating system comes with a system level Python distribution installed (e.g., most flavors of Linux include this), we recommend you follow these instructions to install the standalone Miniconda Python distribution. System level distributions are often far out of date and are difficult to upgrade without breaking the entire operating system.

Miniconda uses the same core packages as Anaconda but strips out many extra packages that may not be needed. If you do need missing components later, it is straightforward to add them individually. Please note that the various Python tools we will use for the workshop will install several packages, which, together with some reference data, take about 13 GB of disk space. Please consider this before deciding to use Python (as opposed to the Google Colaboratory) for the hands-on sessions.

For a Miniconda overview, see: <https://docs.conda.io/en/latest/miniconda.html>.

Download Miniconda: <https://docs.conda.io/en/latest/miniconda.html#installing> – scroll down to find operating system installers and select the **Python 3.9** version.

Note: If you already have Python installed on your system but would like to install the new packages in their own virtual environment, you can choose to install Python in a custom working environment (see below for custom working environment instructions).

Windows MacOS

The Windows and Mac versions of Miniconda are installed by double-clicking the downloaded installer. Follow the prompts to install the Python distribution.

You will interact with Python from a terminal prompt. On macOS you can reach the terminal from your Applications/Utilities directory. On Windows, use the search feature next to the start menu to search for Miniconda. From there open the Powershell to access the command line.

Linux

The 3.9.X Linux version is installed by running the 'sh' script from the command line in a terminal:

```
bash Miniconda3-py39_XXXXX.sh
```

The filename is what you have downloaded above.

After accepting the license agreement, you will be asked for a location into which to install the distribution (e.g., the default `~username/miniconda3`). All of the Python packages we will be using will be installed under the directory you specified. When you are asked "Do you wish the installer to initialize Anaconda3 by running `conda init`?", answer "yes". This will add code to your `.bashrc` file which initializes Miniconda Python whenever you open a new terminal window.

Now close the terminal and open another, so that the new `.bashrc` takes effect.

Update to the Latest Version

Next update the core conda package to the latest version within the 3.9 branch by typing:

```
conda update conda
```

Answer "y" when asked if you wish to update, and wait for the update to finish.

If you would like to update *all* of the packages that come with Miniconda to their latest versions, type:

```
conda update --all
```

Note that this will take some time. This step is optional, since packages can also be updated individually as needed.

You now have a working Python distribution installed in your account.

Custom Working Environment (*Optional*)

If you are developing code or running a program that requires you to install custom packages, it can be useful to create a virtual environment dedicated to that project. This environment allows you to create local installations of your Python environment for specific purposes; if you do something undesirable in a virtual environment, you can delete it and start over, without having to reinstall conda.

Note: This step is optional for the Sagan Workshop. It can be useful for users who are already using Python for other projects, and want to keep the workshop materials separated out.

To create a virtual Python3 environment for use with the Sagan Workshop, type the following:
`conda create --name sagan python=3.9.7`

This creates a directory `~username/miniconda3/envs/sagan` (C:\Users\username\miniconda3\envs\sagan under Windows), which contains a local copy of your Python directory. You can modify the Python environment in here without risk of damaging the base install.

To ‘activate’ your custom environment, type:

```
conda activate sagan
```

Note: You may have to initialize your shell first before this command can be run. To do this,

type `conda init <Shell_name>`

where `<Shell_name>` should be either `bash`, `fish`, `tcsh`, `xonsh`, `zsh`, or `powershell`, then close and restart your shell.

Once you have activated your custom environment, the beginning of your terminal prompt should change to `[sagan]`, and any packages that you install now will end up in `~/miniconda3/envs/sagan`. Note that each new terminal you open needs to have the activate command run on it.

To ‘deactivate’ a custom environment, type:

```
conda deactivate or conda activate
```

(with no environment specified). That will return your terminal to the base environment.

Install wget for Windows

Windows users will need to install `wget` to download data needed for the hands-on sessions within the notebooks. `Wget` or `Curl` may also be used outside of the notebooks to download directly.

```
conda install wget
```

Python Jupyter Notebooks

Jupyter notebooks are a handy, interactive way to work through programming problems in a GUI environment in your web browser. They allow you to easily document your work (just like a real lab notebook) and visualize the results of each step in a computer program. The activities for the Sagan Workshop are in the form of Jupyter notebooks (files ending in ‘ipynb’). Please read the following instructions if you are not familiar with them.

If you installed miniconda, run the following to add the Jupyter notebook package:

```
conda install jupyter
```

You can start running a Jupyter notebook by typing the following command in a terminal window or Powershell in Windows (you should type this in the directory where the Jupyter notebook is located, or any directory above it):

```
jupyter notebook
```

This will launch the Jupyter kernel and load an interface in your local web browser where you can see the content of the directory where you started Jupyter. To open a Jupyter notebook, simply click on the file (it ends in ‘ipynb’), and the notebook will open up in a new browser window. To execute code in a Jupyter notebook cell, type CTRL-Enter (or Shift-Enter). Note that Jupyter notebooks are automatically saved, so it is helpful to create a backup copy of any notebook before working with it.

When you are done, click on the Jupyter icon  in the upper left. This will return you to the directory interface, where you can load additional notebooks.

To close a Jupyter notebook, in the directory interface select the notebook and click on “Shut-down”; close the browser window. Then, if you launched the notebook from a terminal window, press Control-C twice in that window to shut down the notebook server. Another option to shut down the server is to click on “Quit” in the top right corner of the directory interface page and close the browser window.

SSW2023 Workshop Packages

Before installing any of the workshop packages, create a directory on your computer for the hands-on sessions, and then proceed with the installation commands from this hands-on sessions' directory.

Hands-on Session I & II

1) Install Eureka! *(from the command line):*

This will install Eureka! and several packages it depends on:

```
pip install eureka[jwst]@git+https://github.com/kevin218/Eureka.git@dev/tjb
```

It is also important to setup the CRDS environment variables as described at

<https://eurekadocs.readthedocs.io/en/latest/installation.html#crds-environment-variables>

If you encounter issues with the jwst requirement, follow the instructions at

<https://eurekadocs.readthedocs.io/en/latest/faq.html#faq-install>

Users with Apple M1/M2 processors should read the relevant FAQ at

<https://eurekadocs.readthedocs.io/en/latest/faq.html#installation-issues-with-m1-processors>

For more details, including different installation options, see

<https://eurekadocs.readthedocs.io/en/latest/installation.html>

2) Run the Setup notebook for Session I & II to download data (see below for the link.)

Hands-on Session III

1) Install PICASO *(from the command line; note, will use 1.2GB of space):*

```
pip install picaso
```

For more details, see <https://natashabatalha.github.io/picaso/installation.html>.

2) Run the Setup notebook for Session III to download data (see below for the link.)

Hands-on Session IV

Please note that the installation of petitRADTRANS (pRT) also requires C and Fortran compilers. While installing pRT *can* go very smoothly, depending on your operating system and on whether you have an Apple chip (M1 or M2) in your Mac, things can become more complicated. pRT has been developed using a Mac (Intel chip) and Linux, but users have contributed various instructions for installing pRT on other operating systems (Mac with Apple Chips, Windows). You will find detailed descriptions of the installation process on the [pRT website](#). With around 1000 registered participants, we ask for your understanding that we cannot help you installing pRT if you run into problems during the workshop. Therefore, if installing pRT on your machine does not work, please use the Google Colab notebooks instead.

1) Install petitRADTRANS (pRT) (from the command line):

```
pip install pymultinest

git clone https://github.com/JohannesBuchner/MultiNest
cd MultiNest/build
cmake ..
make
```

After installing MultiNest, you need to copy the library files from MultiNest/lib to the lib directory of your Python installation. You can type

```
which python
```

to see the Python directory structure. The output is, e.g.,
/YOUR_DIRECTORIES/Python/miniconda3/envs/sagan/bin/python

In this case your Python library directory is

```
/YOUR_DIRECTORIES/Python/miniconda3/envs/sagan/lib
```

Copy the files from MultiNest/lib to this directory.

```
pip install petitRADTRANS
```

For more details about installing MultiNest, see <https://johannesbuchner.github.io/PyMultiNest/install.html#building-the-libraries>

For more details about installing pRT, see the installation instructions given on the [pRT website](#).

2) Run the Setup notebook for Session IV to download data (see below for the link.)

SSW 2023 Notebooks


[NOTE: the notebook links have been updated to their permanent archive location]

For the notebooks below, simply save them from your browser window by using the menu option File → Download → Download .ipynb.

Another option is to download all notebooks:

Tar format: https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_notebooks.tar.gz

Zip format: https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_notebooks.zip

 Note: Windows users may need to add “`--no-certificate-check`” to the `wget` lines in Python cells in the Setup notebooks.

Hands-on Session I: Reducing JWST Data: From Raw Data to Light Curves

SSW2023_JWST_MIRI_NIRSpec_Setup.ipynb

(Note: needs to be run just **once** and applies to Hands-on Sessions I & II)

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_JWST_MIRI_NIRSpec_Setup.ipynb

SSW2023_Workshop1_MIRI_Handout.ipynb

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_Workshop1_MIRI_Handout.ipynb

SSW2023_Workshop1_NIRSpec_Handout.ipynb

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_Workshop1_NIR-Spec_Handout.ipynb

Hands-on Session II: Fitting JWST Data: From Light Curves to Planet Spectra

SSW2023_JWST_MIRI_NIRSpec_Setup.ipynb

(Note: skip this step if you've already run it for Hands-on Session I)

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_JWST_MIRI_NIRSpec_Setup.ipynb

SSW2023_Workshop2_MIRI_Handout.ipynb

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_Workshop2_MIRI_Handout.ipynb

SSW2023_Workshop2_NIRSpec_Handout.ipynb

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_Workshop2_NIR-Spec_Handout.ipynb

Hands-on Session III: Forward Modeling with PICASO

SSW2023_Forward_Model_Picaso_Setup.ipynb

(Note: needs to be run just once and applies to Hands-on Session III)

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_Forward_Model_Picaso_Setup.ipynb

SSW2023_Forward_Model_Picaso.ipynb

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_Forward_Model_Picaso.ipynb

Hands-on Session IV: Retrievals using petitRADTRANS

SSW2023_Retrievals_using_petitRADTRANS_Setup.ipynb

(note: needs to be run just once and applies to Hands-on Session IV)

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_Retrievals_using_petitRADTRANS_Setup.ipynb

SSW2023_petitRADTRANS_intro_exercises.ipynb

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_petitRADTRANS_intro_exercises.ipynb

SSW2023_petitRADTRANS_retrieval_exercises.ipynb

https://catcopy.ipac.caltech.edu/ssw2023/hands-on/SSW2023_petitRADTRANS_retrieval_exercises.ipynb