

Sagan Workshop 2020: Getting Started with Python

The hand-on activities we will use during the workshop are based around Jupyter Notebooks using software written in Python3. If you are unfamiliar with these tools, the guide below should help you set them up on your personal computer and provide an overview of their capabilities. *You should setup your Python distribution in advance of the workshop; if you need assistance, reach out via the Slack channel!* In this document, we use the `Courier New font` to indicate a command you type into the command prompt.

The Python Distribution

You first need a working copy of Python3. 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 Anaconda Python distribution. System level distributions are often far out of date, and are difficult to upgrade without breaking the entire operating system.

You can download the free Anaconda here: <https://www.anaconda.com/products/individual>

Versions exist for Windows, Mac, and Linux. *Choose the Python 3.7 download appropriate for your operating system.* Note that the Individual Edition is free, while the Team and Enterprise distributions require license agreements. For the Sagan Workshop, and most data analysis you would do for independent research, the Individual Edition is sufficient.

These downloads are between 300-600 MB, depending on operating system. If your internet is limited in bandwidth, you can utilize the Miniconda distribution, which is much smaller (~50 MB) at <https://docs.conda.io/en/latest/miniconda.html> . This gives you the same core packages as Anaconda, but strips out many extra packages that may not be needed. If you do need missing components later on, it is straightforward to add them individually (see page 3 for a list of the main packages you would have to install).

Windows  **MacOS** 

The Windows and Mac versions of Anaconda and Miniconda are installed in the normal way by double-clicking the downloaded installer. Follow the prompts to install the Python distribution. Be patient as the installation can take as much as 15-20 min to complete. 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 Anaconda Navigator. From there open the Powershell to access the command line.

Linux

The Linux version is installed by running the 'sh' script from the command line in a terminal:
`bash Anaconda3-2020.02-Linux-x86_64.sh`

After accepting the license agreement, you will be asked for a location into which to install the distribution (e.g. the default `/home/username/anaconda3`). All of the Python packages we will be using will be installed under the directory you specified. This will take quite some time to complete. 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 Anaconda 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 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 Anaconda 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.

Install Packages for the Workshop

We will use a few additional Python packages that are not included in the Anaconda distribution. You should install them by typing:

```
conda install -c astropy astroquery
pip install datetime
pip install corner
pip install --extra-index-url https://dace.unige.ch/api python-dace-client
pip install --extra-index-url https://obswww.unige.ch/~delisle spleaf
pip install --extra-index-url https://obswww.unige.ch/~delisle kepderiv
```

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=3.7`

This creates a directory `~/anaconda3/envs/sagan` (C:\Users\username\anaconda3\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
```

The beginning of your terminal prompt should change to `[sagan]`, and any packages that you install now will end up in `~/anaconda3/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.

By default, conda environments contain only a minimal set of packages. If you are working in an environment, you will need to install packages as necessary for the workshop; here are the main ones (in addition to the ones listed on page 2):

```
conda activate sagan
conda install numpy
conda install scipy
conda install astropy
conda install pandas
conda install matplotlib
conda install jupyter
```

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, so if you are not familiar with them, read the following instructions. *Note: We are still using the classic Notebook interface, not the new JupyterLab interface.*

Jupyter comes included with the full Anaconda distribution.

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

```
conda install -c conda-forge notebook
```

You can start running a Jupyter notebook with:

```
jupyter notebook
```



Note: Windows users can use Anaconda Navigator to access the Jupyter notebook interface

This will launch the Jupyter kernel and load an interface in your local web browser. If you are not familiar with Python, you should work through the tutorial: *Sagan2020PythonIntro.ipynb*

This is a Jupyter notebook, and you can open it from the Jupyter interface in your browser.

Work your way through the notebook. Interactive code is indicated by this:

Example

```
message = "Hello Python world!"  
print(message)
```

You can modify the commands in the shaded box. To re-run the code, type CTRL-Enter (or Shift-Enter).

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.