

Everything is better with friends: Executing SAS[®] code in Python scripts with SASPy

PYTHON CHEATSHEET (FOR SAS PROGRAMMERS)

Python is a versatile programming language with syntax resembling DATA steps in SAS, but with the following important differences.

Capitalization matters

`print('Hello, World!')` is not the same as `PRINT('Hello, World!')`

Semicolons are only used when multiple statements are on the same line

`message = 'Hello'`
`print(message)` is the same as `message = 'Hello'; print(message)`

There are multiple, (mostly) interchangeable quoting styles

`print('A')` is the same as `print("A")`, `print(''A'')`, or `print("""A""")`

Assignment is different than equality testing

Assignment: `fireworks = 'Yes!'` Equality Testing: `fireworks == 'Yes!'`

White space matters (and is used to determine scope)

`if fireworks == 'Yes!':`
`print('💣')` would produce an error with indentation removed

REPLICATING HANDS-ON WORKSHOP EXAMPLES

Step 1: Download and install the freely available SAS University Edition. We recommend using VirtualBox:
<https://support.sas.com/software/products/university-edition/docs/en/SASUniversityEditionQuickStartVirtualBox.pdf>

Step 2: Start SAS University Edition, and access JupyterLab within a web browser:
https://support.sas.com/software/products/university-edition/faq/jn_runvirtualbox.htm

Step 3: Download example file **SGF2019-HOW-Everything_Is_Better_With_Friends-examples.ipynb** from
<https://github.com/saspy-bffs/sgf-2019-how>

Step 4: Load example file into JupyterLab: <https://jupyterlab.readthedocs.io/en/stable/user/files.html>

Step 5: Follow instructions in example file.

REPLICATING HANDS-ON WORKSHOP DEMOS

Because Python¹ is community-driven, there is immense flexibility in how it can be used. The following is a short guide for setting up SASPy with the popular IDE PyCharm². These instructions were developed using Windows 10 with SAS 9.4, Java SE version 8 update 201, Python 3.7.3, Git 2.21.0, and PyCharm Professional Edition 2019.1.1 installed. All default installation options are recommended.

Step 1: Setup development environment.

- Download and install Java SE (to be used by SASPy to invoke SAS) from <https://www.java.com/>
- Download and install Python (i.e., the CPython implementation) from <https://www.python.org/>
- Download and install Git (to be used by PyCharm's GitHub integration) from <https://git-scm.com/>
- Download and install PyCharm from <https://www.jetbrains.com/pycharm/>.

Step 2: Setup project in PyCharm.

- Start PyCharm, and when prompted, choose **Check out from Version Control** → **Git**, enter URL <https://github.com/saspy-bffs/sgf-2019-how>, choose a directory to copy the files to, and click **Clone**.
- Use the menu commands **File** → **Settings** → **Project: sgf-2019-how** → **Project Interpreter**. Then click the **Gear Icon** (⚙) in the upper-right corner of the dialog box and select **Add...** This should prompt you to create a new virtual environment³ as a subfolder named *venv* in your project folder. Once setup, click **OK** and, once processing has finished, click **OK** again to exit the dialog box.
- Use the menu command **View** → **Tool Windows** → **Terminal** to open a terminal window. (Alternatively, click **Terminal** at the bottom of the PyCharm window.) Then type the following at the command prompt and press Enter: **pip install -r requirements.txt**

Use the project-navigation area in the left-hand panel to open the file *sascfg_personal-example.py* (i.e., double-click its name), and copy its contents to the system clipboard. Then use the menu command **File** → **New** → **Python File** to create a new file named *sascfg_personal.py*, paste the contents of the system clipboard into it, and update to match your SAS installation setup per the instructions at <https://sassoftware.github.io/saspy/install.html>.

Step 3: Run the example file.

- Use the menu command **Run** → **Run ...** → **SGF2019-HOW-Everything_Is_Better_With_Friends-examples.py**, and then watch the output scroll by in the Run portion of bottom panel. (Alternatively, open *SGF2019-HOW-Everything_Is_Better_With_Friends-examples.py* by double-clicking its name in the project-navigation area in the left-hand panel, right-click anywhere inside the code editor window, and choose the command **Run 'SGF2019-HOW-Everything...'**)

Step 4: Repeat with a full-application example using the popular Python web framework Flask.

- Use the menu command **VCS** → **Checkout from version control** → **Git** to create a new project from <https://github.com/saspy-bffs/dataset-explorer> and repeat Step 2(b-d). Then run *app.py*, visit the URL <http://127.0.0.1:8000/> in a web browser, and follow the instructions. (If SAS was installed with default options, try using the directory **C:\Program Files\SASHome\SASFoundation\9.4\core\sashelp**.)

¹ When most people talk about the Python language, they mean the C-based CPython reference implementation at <https://github.com/python/cpython>. Other implementations of the Python language specification include Java-based Jython (<https://www.jython.org/>) and .NET-based Iron Python (<https://ironpython.net/>). The CPython implementation can also be installed as part of the data-science oriented Anaconda distribution (<https://www.anaconda.com/distribution/>).

² You can choose between PyCharm Community Edition, which is free and open-source, and PyCharm Professional Edition, which requires a license after an initial demo period.

³ A virtual environment (aka *venv*) is essentially a completely separate installation of Python, which is cloned from the version of Python installed as Step 1(b). It's considered best practice to create a new *venv* for each project in order to keep its dependencies isolated from other projects having separate dependencies.