

9 Coding in Python

Topics: Python data types, variables, functions, importing libraries, connecting code to math and physical science.

9.1 Motivation and approach

Translating mathematical concepts into code is an amazingly useful skill for anything from data analysis and making plots to figures to simulating the behavior of a protein interacting with an enzyme. In this section we will go over an introduction to programming in Python followed by several detailed examples of how you can use these coding concepts to solve interesting problems. As coding is best learned when you are able to try it yourself, the coding notes are provided in python notebooks linked below. Follow the instructions in Appendix A to download copies of these notebooks to your UC Berkeley-provided JupyterHub accounts.

[Introduction to Python](#)

[Scientific Python](#)

A Jupyter Notebooks

A.1 Introduction

Jupyter Notebook is a useful tool that allows you to create an online notebook which can contain writing, equations, pictures, and code all in one place. Jupyter notebooks even allow you to run python code without installing any programs on your personal computer. In this tutorial we'll go through setting up your first Jupyter notebook, as well as some useful tips and tricks to make the most out of this powerful tool.

A.2 Creating a Notebook

Jupyter notebooks are available for free online, but have limited capabilities without a subscription. Fortunately for us, Berkeley provides everyone with a subscription so long as you log in through their interface [here](#).

Once you've logged in, you'll be presented with your home page. Here you can find your previous notebooks, make a new notebook, or even upload another notebook that is saved to your computer. Let's start by making our own python Jupyter notebook. Click "new" in the top right corner and select a Python 3 notebook.

A.3 Working in a Notebook

In your new notebook, you'll be presented with a blank cell. Cells are where your content goes, and come in a few varieties. Code cells contain code, in this particular case python code because this is a python notebook, and are also able to execute this code. Markdown cells contain text and equations, and are even able to compile typesetting from your favorite typesetting language, such as LaTeX. Let's start by practicing with code cells.

As stated before, code cells are where you will write and execute code. Let's start by writing a very simple python code. Putting `print('Hello World!')` into the cell (without the double quotations) will write a simple program which will display "Hello World!" when run. To run the code, simply press the play button on the toolbar or press `ctrl + enter`. You should now see the output printed below the cell! You also may have noticed the small circle in the top right corner. This circle tells you about the kernel, which is what actually running the code. Hovering over the kernel will tell you what the kernel is doing. Sometimes you may run into issues where the kernel disconnects or where you want to delete all the memory. To fix this, it is often helpful to restart the kernel. To do this, simply click the Kernel dropdown menu and click restart.

Now that you've run your code, you may find that you want to delete the output so that you can run your code again. To clear output, click the cells dropdown menu and select either Current Output \rightarrow clear to clear output in only the selected cell or All Output \rightarrow clear to clear output from all cells. Now let's create a new markdown cell and write an equation. To create a new cell, click the insert dropdown and then cell below, or just type `shift + enter` (note that this will also run the current cell).

By default, our new cell is a code cell. To change the cell to a markdown cell, click the dropdown on the toolbar that currently says code and select markdown. Now that we're in a markdown cell, we can type equations or text like in a typesetting program. For instance, if we type `"$y=\frac{1}{3}x+b$"` into our cell and run it, an equation will appear!

Keyboard shortcuts can save you a lot of time in Jupyter Notebooks, we suggest you check out this list: <https://www.earthdatascience.org/courses/intro-to-earth-data-science/open-reproducible-science/jupyter-python/jupyter-notebook-shortcuts/>. You can access a list of all keyboard shortcuts by hitting `esc + h`.

A.4 Closing Remarks

Jupyter Notebooks is a powerful tool that allows you to keep all your code, writing, and equations in one convenient place. It makes sharing and running code more accessible, and saves the hassle of installing a programming language onto your own personal device. While this tutorial was merely a bare minimum to create your first notebook and familiarize yourself with some of the essential tools, there are many more features, shortcuts, and hotkeys to take advantage of, so I encourage you all to search on the internet for more tools if you are interested in making extensive use of Jupyter Notebooks!