

Jupyter: Beyond Normal Python

There are many options for development environments for Python, and I'm often asked which one I use in my own work. My answer sometimes surprises people: my preferred environment is [IPython](http://ipython.org/) (<http://ipython.org/>) plus a text editor (in my case, Emacs or VSCode depending on my mood). Jupyter got its start as the IPython shell, which was created in 2001 by Fernando Perez as an enhanced Python interpreter and has since grown into a project aiming to provide, in Perez's words, "Tools for the entire life cycle of research computing." If Python is the engine of our data science task, you might think of Jupyter as the interactive control panel.

As well as being a useful interactive interface to Python, Jupyter also provides a number of useful syntactic additions to the language; we'll cover the most useful of these additions here. Perhaps the most familiar interface provided by the Jupyter project is the Jupyter Notebook, a browser-based environment that is useful for development, collaboration, sharing, and even publication of data science results. As an example of the usefulness of the notebook format, look no further than the page you are reading: the entire manuscript for this book was composed as a set of Jupyter notebooks.

This part of the book will start by stepping through some of the Jupyter and IPython features that are useful to the practice of data science, focusing especially on the syntax they offer beyond the standard features of Python. Next, we will go into a bit more depth on some of the more useful *magic commands* that can speed up common tasks in creating and using data science code. Finally, we will touch on some of the features of the notebook that make it useful for understanding data and sharing results.