What are Jupyter Notebooks?

Welcome to this lesson on using Jupyter notebooks. The notebook is a web application that allows you to combine explanatory text, math equations, code, and visualizations all in one easily sharable document. For example, here's one of my favorite notebooks shared recently, binary black hole signals in LIGO open data detected by the LIGO experiment. You could download the data, run the code in the notebook, and repeat the analysis, in effect detecting the gravitational waves yourself! You can view a few more tutorial notebooks at Gravitational Wave Open Science Center homepage.

Notebooks have quickly become an essential tool when working with data. You'll find them being used for data cleaning and exploration, visualization, machine learning, and big data analysis. Here's an example notebook I made for my personal blog that shows off many of the features of notebooks. Typically you'd be doing this work in a terminal, either the normal Python shell or with IPython. Your visualizations would be in separate windows, any documentation would be in separate documents, along with various scripts for functions and classes. However, with notebooks, all of these are in one place and easily read together.

Notebooks are also rendered automatically on GitHub. It's a great feature that lets you easily share your work. There is also http://nbviewer.jupyter.org/ that renders the notebooks from your GitHub repo or from notebooks stored elsewhere.

Literate Programming

Notebooks are a form of literate programming proposed by Donald Knuth in 1984. With literate programming, the documentation is written as a narrative alongside the code instead of sitting off by its own. In Donald Knuth's words,

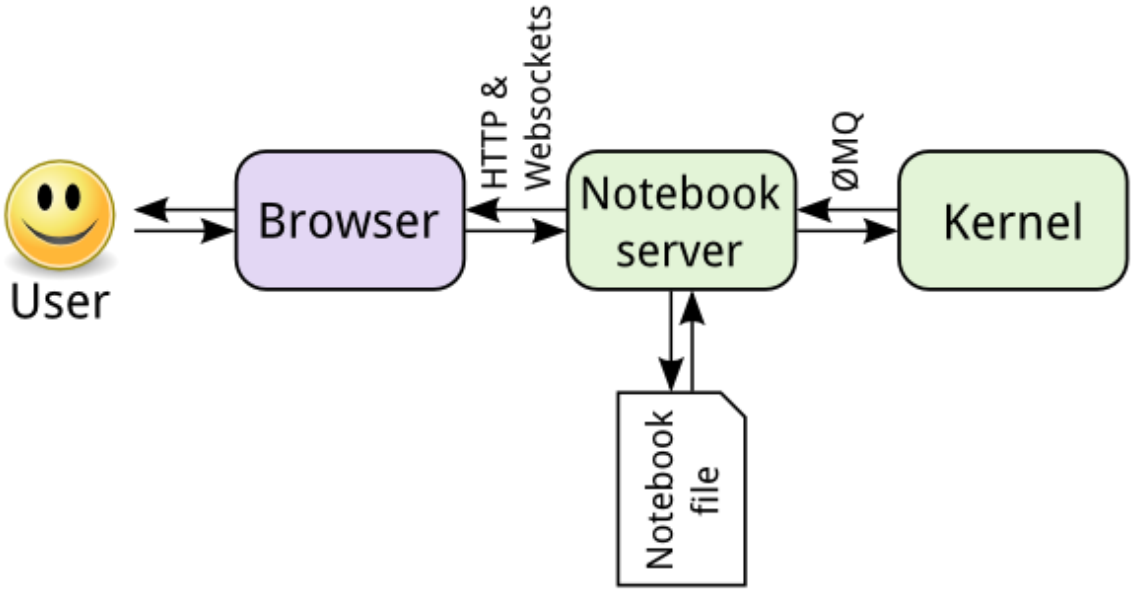
Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

After all, code is written for humans, not for computers. Notebooks provide exactly this capability. You are able to write documentation as narrative text, along with code. This is not only useful for the people reading your notebooks, but for your future self coming back to the analysis.

Just a small aside: recently, this idea of literate programming has been extended to a whole programming language, Eve.

How Notebooks Work

Jupyter notebooks grew out of the IPython project started by Fernando Perez. IPython is an interactive shell, similar to the normal Python shell but with great features like syntax highlighting and code completion. Originally, notebooks worked by sending messages from the web app (the notebook you see in the browser) to an IPython kernel (an IPython application running in the background). The kernel executed the code, then sent it back to the notebook. The current architecture is similar, drawn out below.



From Jupyter documentation

The central point is the notebook server. You connect to the server through your browser and the notebook is rendered as a web app. Code you write in the web app is sent through the server to the kernel. The kernel runs the code and sends it back to the server, then any output is rendered back in the browser. When you save the notebook, it is written to the server as a JSON file with a .ipynb file extension.

The great part of this architecture is that the kernel doesn't need to run Python. Since the notebook and the kernel are separate, code in any language can be sent between them. For example, two of the earlier non-Python kernels were for the R and Julia languages. With an R kernel, code written in R will be sent to the R kernel where it is executed, exactly the same as Python code running on a Python kernel. IPython notebooks were renamed because notebooks became language agnostic. The new name **Jupyter** comes from the combination of **Ju**lia, **Pyt**hon, and **R**. If you're interested, here's a list of available kernels.

Another benefit is that the server can be run anywhere and accessed via the internet. Typically you'll be running the server on your own machine where all your data and notebook files are stored. But, you could also set up a server on a remote machine or cloud instance like Amazon's EC2. Then, you can access the notebooks in your browser from anywhere in the world.