Introduction2Jupyter

April 12, 2024

1 Introduction to Jupyter

1.1 What is Jupyter Notebook?

A Jupyter Notebook is a web browser based **interactive computing environment** that enables users to create documents that include code to be executed, results from the executed code such as plots and images,and finally also an additional documentation in form of markdown text including equations in LaTeX.

These documents provide a **complete and self-contained record of a computation** that can be converted to various formats and shared with others using email, version control systems (like git/GitHub) or nbviewer.jupyter.org.

The Jupyter Notebook combines three components:

- **Notebook editor**: an interactive application for writing and running code interactively and editing notebook documents. If you run Jupyter on desktop, you will be using Jupyter's web application.

- **Kernels**: Separate processes started by Jupyter on your server, that runs users' code in a given language and returns output back to the notebook web application. The kernel also handles things like computations for interactive widgets, tab completion and introspection.

- **Notebook documents**: Self-contained documents that contain a representation of all content visible in the notebook editor, including inputs and outputs of the computations, markdown text, equations, images, and rich media representations of objects. Each notebook document has its own kernel.

1.2 Notebook editor

The Notebook editor is web application running in your browse. It enables you to

- **Edit code** in individual cell
- **Run code** in individuall cells in arbitrary oder and display results of the computation in various formats (HTML, LaTeX, PNG, SVG, PDF)
- Create and use **interactive JavaScript widgets**, which bind interactive user interface controls and visualizations to reactive kernel side computations.
- Add **documentation text** using Markdown markup language, including LaTeX equations

1.3 Kernels

The Jupyter notebook is not bound to any specific programming language, but can be used for almost any type of language. Each Jupyter notebook starts a server application that is connected to a kernel that runs the code in the notebook. This kernel is dedicated to a specific programming language. Thus installing different kernels 100+ languages allows you to execute code in **Python**, **Julia**, **R**, **Ruby**, **Haskell**, **Scala**, and many others.

Yet, the default kernel runs Python code. The notebook provides a simple way for users to pick which of these kernels is used for a given notebook. Each of these kernels communicate with the notebook editor using JSON over ZeroMQ/WebSockets message protocol that is described here. Most users don't need to know about these details, but it helps to understand that "kernels run code".

1.4 Notebook documents

Notebook documents, or notebooks, contain the **inputs and outputs** of an interactive session as well as **documentation text** that accompanies the code but is not meant for execution.

A notebook is just a **file on your server's filesystem with a `.ipynb` extension**. This allows you to share your notebook easily.

Notebooks consist of a **linear sequence of cells**. There are three basic cell types:

- **Code cells:** Input and output of live code that is run in the kernel.
- **Markdown cells:** Narrative text with embedded LaTeX equations.
- **Raw cells:** Unformatted text that is included, without modification, when notebooks are converted to different formats using `nbconvert`.

Internally, notebook documents are **JSON** **text files** with **binary data encoded in base64**. This allows them to be **read and manipulated programmatically** by any programming language.

Notebooks can be exported to different static formats including HTML, reStructuredText, LaTeX, PDF, and slide shows (reveal.js) using Jupyter's `nbconvert` utility.

Furthermore, any notebook document available from a **public URL on or GitHub can be shared** via nbviewer. This service loads the notebook document from the URL and renders it as a static web page. The resulting web page may thus be shared with others **without their needing to install the Jupyter Notebook**.