

Teaching and Learning with Jupyter

Lorena A. Barba et al.

2018-11-30

Contents

1	Prerequisites	5
2	Introduction	7
2.1	Acknowledgements	8
3	About the Authors	9
3.1	Project Lead	9
3.2	Authors at the sprint	9
4	Glossary	13

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")  
# or the development version  
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading #.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.name/tinytex/>.

Chapter 2

Introduction

Project Jupyter is a broad collaboration that develops open-source tools for interactive and exploratory computing. The tools include: IPython, the Jupyter Notebook, Jupyter Hub, and an ecosystem of extensions contributed by a large community. Jupyter Notebook exploded in popularity since late 2014, fueled by its adoption as the favorite environment for doing data science. It has also grown as a platform to use in the classroom, to develop teaching materials, to share lessons and tutorials, and more. Notebooks are documents containing text narratives, combined with executable code (many languages are supported) and the output of that code. This marriage of content and code makes for a powerful new form of data-based communication. Educators everywhere are adopting Jupyter for teaching.

This handbook is for any educator teaching a topic that includes data analysis or computation to support the learning—not just courses in engineering or science, but also data journalism, business and quantitative economics, data-based decision sciences and policy, quantitative health sciences, and others. It aims to give an entry point, and a broad overview of Jupyter in education. Whether you are already using Jupyter to teach, you have found learning materials built on Jupyter that piqued your curiosity, or have never heard of Jupyter, the material in this open book can help you empower your teaching with this new technology.

Educators newly adopting Jupyter can be overwhelmed by having to navigate the ecosystem of tools and content. They could study many examples, or consume myriad blog posts and talk videos to distill the patterns of good practices and technical solutions to best serve their students. Several early adopters, having much experience to share, decided to begin collecting this know-how, and sharing open documentation about using Jupyter for teaching and learning.

The Jupyter Community Workshop in DC (November 2018) began that process, with a book sprint aimed at producing the first version of this handbook. The collaboratively written book consolidates explanations and examples covering key topics, including: what is Jupyter, how to try Jupyter, sharing notebooks with students, locally installing Jupyter, cloud offerings, finding example notebooks, writing lessons in Jupyter, making collections for a course, exporting to other formats with nbconvert, writing textbooks with Jupyter, using Binder, JupyterHub, making assignments and auto-grading, making online courses, teaching with Jupyter in the classroom, active learning and flipped learning pedagogies with Jupyter, guiding learners to create their own content in Jupyter, and more. This open handbook will grow to encompass all you need to know about Jupyter in Teaching and Learning.

If you find these materials helpful or inspiring, give us a shout-out on Twitter using #Jupyter4Edu. We hope you do!

2.1 Acknowledgements

The book sprint was held at the George Washington University in Washington, DC, on 28–30 November 2018, and organized by Lorena A. Barba. Funding to support the logistics and travel of all participants was possible thanks to a grant from Bloomberg to Project Jupyter, and managed by NumFOCUS. The group was fêted at a reception sponsored by Leidos. Participants traveled from all over the country and volunteered their precious time and hard work to give this work to the Jupyter community, with a heartfelt sense of gratitude to all the contributors to the software projects we love and depend on. Thank you!

GitHub repository for this book: <https://github.com/jupyter4edu/jupyter-edu-book>

Chapter 3

About the Authors

3.1 Project Lead

3.1.1 Lorena A. Barba

- George Washington University
- labarba@email.gwu.edu
- @LorenaABarba

Lorena A. Barba is Associate Professor of Mechanical and Aerospace Engineering at the George Washington University. She adopted Jupyter in 2013 and since then used it in every course she teaches. Her open course materials are well known and used by thousands of learners: CFD Python and Numerical MOOC are the best examples.

3.2 Authors at the sprint

3.2.1 Lecia J. Barker

- University of Colorado Boulder
- lecia.barker@colorado.edu
- @leciab

Lecia Barker is an Associate Professor and Associate Chair of Undergraduate Studies in the Department of Information Science at the University of Colorado Boulder. She is also a Senior Research Scientist for the National Center for Women & IT. Her research group is studying the diffusion and adoption of teaching practices in undergraduate computer science. Lecia holds a Ph.D. in Communication from CU Boulder and an MBA in Marketing from San Diego State University.

3.2.2 Douglas Blank

- Bryn Mawr College
- dblank@brynmawr.edu
- @dougblank

Douglas Blank is Associate Professor in the Department of Computer Science at Bryn Mawr College, a small, all-women's college outside of Philadelphia, PA, USA. He has a joint Ph.D. in Cognitive Science and

Computer Science from Indiana University, Bloomington. For over 20 years, Douglas has taught all levels of Computer Science. For the last 4 years, he has used Jupyter notebooks exclusively in the classroom. Douglas has published in the areas of Computer Science Education, Robotics, Artificial Intelligence, and Deep Learning. He is on the advisory board of Engage-CSEdu.org, a joint project between Google and the National Center for Women and Information Technology (NCWIT). Douglas also writes text and code at his website douglasblank.com.

3.2.3 Jed Brown

- University of Colorado Boulder
- jed@jedbrown.org
- @five9a2

Jed Brown is an Assistant Professor of Computer Science at the University of Colorado Boulder. He has been teaching numerical and scientific computing courses using Jupyter Notebook and nbgrader for three years, and leads a research group that develops computational methods and community software for computational science.

3.2.4 Allen Downey

- Olin College
- downey@allendowney.com
- @AllenDowney

Allen Downey is a professor of Computer Science at Olin College and the author of a series of open-source textbooks related to software and data science, including *Think Python*, *Think Bayes*, and *Think Complexity*, published by O'Reilly Media. These books, and the classes based on them, use Jupyter notebooks extensively. Prof Downey holds a Ph.D. in computer science from U.C. Berkeley, and M.S. and B.S. degrees from MIT.

3.2.5 Tim George

- Project Jupyter
- tgeorgeux@gmail.com

Timothy George is the Lead UI/UX Designer for Project Jupyter, focusing primarily on JupyterLab. In addition to his formal duties, Tim is also in working with Jupyter on design strategy, future products, governance, diversity and inclusion. He studied HCI at UC Irvine's Donald Bren School of Informatics and Computer Science where he received a Master's Degree.

3.2.6 Lindsey Heagy

- University of California Berkeley
- lindseyheagy@gmail.com
- @lindsey_jh

Lindsey Heagy is a Postdoctoral Researcher at the University of California Berkeley working on Project Jupyter and Jupyter in the geosciences. She recently completed her PhD at the University of British Columbia in geophysics. She is a project leader of GeoSci.xyz, an effort to build collaborative, interactive, web-based textbooks in the geosciences, and a core contributor to SimPEG, an open source framework for geophysical simulation and inversions. The GeoSci.xyz project relies heavily on Jupyter for making the content come to life.

3.2.7 Kyle Mandli

- Columbia University
- kyle.mandli@columbia.edu
- @KyleMandli

Kyle Mandli is an Assistant Professor in the Department of Applied Physics and Applied Mathematics at Columbia University. He has developed a set of openly available course notes centered around Jupyter notebooks and uses Jupyter for homework in conjunction with nbgrader. His other research interests include development of computational methods for coastal hazards such as storm surge and tsunamis.

3.2.8 Jason K. Moore

- University of California, Davis
- jkm@ucdavis.edu
- @moorepants

Jason K. Moore is an Assistant Teaching Professor of Mechanical and Aerospace Engineering at the University of California, Davis. He currently teaches dynamics and mechanical design related courses. He utilizes Jupyter notebooks to teach modeling and simulation and is working on a textbook about Mechanical Vibrations. He is responsible for the Jupyter related features in the LibreTexts project and is also a core developer of the SymPy and PyDy projects which utilizes Jupyter for training workshops, e.g. PyDy Tutorial and SymPy Code Generation Tutorial. Jason has PhD, MSc, and BSc degrees in mechanical engineering from UC Davis and Old Dominion University.

3.2.9 David Lippert

- Leidos

David Lippert is a software engineer at Leidos in Arlington, Virginia. He utilizes Jupyter notebooks primarily for exploratory data analysis and for training and evaluating machine learning algorithms. He has written Jupyter notebooks to create new Dr. Seuss sonnets and to evaluate if the Rotten Tomatoes Tomatometer can be trusted. He has a BA in computer science from Middlebury College.

3.2.10 Kyle Niemeyer

- Oregon State University
- kyle.niemeyer@oregonstate.edu
- @kyleniemeyer

Kyle Niemeyer is an Assistant Professor of Mechanical Engineering in the School of Mechanical, Industrial, and Manufacturing Engineering at Oregon State University. He teaches courses in numerical and analytical methods for solving differential equations as well as gas dynamics, and recently developed a graduate course on software development for engineering research. His research group develops and applies methods for modeling combustion and chemically reacting fluid flows. He is also on the steering committee of the Cantera open-source project for chemical kinetics, thermodynamics, and transport processes.

3.2.11 Ryan Watkins

- George Washington University
- rwatkins@gwu.edu
- @parsingscience

Ryan Watkins is a Professor of Educational Technology at George Washington University in Washington DC. He leads the Human-Technology Collaboration (HTC) PhD program area, and he teaches courses in needs assessment, instructional design, and research methods. Ryan’s research focuses on how people and organizations define and assess needs. He is co-host of Parsing Science, a podcast where researchers share the stories behind their science. He also developed the We Share Science platform for sharing video abstracts of research.

3.2.12 Richard West

- Northeastern University
- R.West@northeastern.edu
- @richardhwest

Richard West is Associate Professor of Chemical Engineering at Northeastern University in Boston. He leads a research group in computational modeling for complex reacting systems like combustion or catalysis. He is a core member of the Cantera open-source project. As well as in an elective on “computational modeling in chemical engineering”, he has integrated Python and Jupyter into core classes on chemical kinetics and reactor design, at both the undergraduate and graduate levels. As part of his NSF CAREER award, he is developing modules to teach students to use Python and SciPy to solve chemical engineering problems.

3.2.13 Elizabeth Wickes

- University of Illinois at Urbana-Champaign
- wickes1@illinois.edu
- @elliewix

Elizabeth Wickes is a Lecturer at the School of Information Sciences at the University of Illinois at Urbana-Champaign. She teaches foundational programming from an information and data sciences perspective, as well as other coursework on open data and reproducibility. Her programming course lectures are written in Jupyter Notebooks and the class is taught via live coding.

3.2.14 Carol Willing

- Cal Poly San Luis Obispo
- willingc@gmail.com
- @WillingCarol

Carol Willing is a Research Software Engineer at Cal Poly San Luis Obispo working full-time on Project Jupyter. She is a Python Software Foundation Fellow and former Director; a Project Jupyter Steering Council member; and a core developer on CPython and Jupyter. Carol has an M.S. in Management from MIT and a B.S.E. in Electrical Engineering from Duke.

3.2.15 Michael Zingale

- Stony Brook University
- Michael.Zingale@stonybrook.edu
- @Michael_Zingale

Michael Zingale is an Associate Professor and computational astrophysicist at Stony Brook University. He has a PhD from University of Chicago (2000). He frequently teaches numerical methods and Python for scientific computing graduate courses, relying on Jupyter notebooks and python for much of the presentation. He is an advocate for open educational resources, as a founder of the Open Astrophysics Bookshelf project where he hosts his *Introduction to Computational Astrophysical Hydrodynamics* text.

Chapter 4

Glossary

Anaconda: a free, open-source package manager, environment manager, Python distribution, and collection of over 1,500+ open source packages including and also Jupyter. <https://www.anaconda.com/what-is-anaconda/>

API (Application Programming Interface): the exact details of interacting with software, usually used by other software.

Binder: a hosted service that allows anyone to launch their own sandboxed notebook environment from a Git repository. <https://mybinder.org>

cell: the area in a Jupyter Notebook where you can enter markdown, or computer code.

cloud, in the: used to describe software or documents hosted on a remote computer accessed over the internet.

CSV (Comma Separated Values): referring to a comma-separated value file. A plain-text file format such that each line is a list of data separated by commas.

execute: technical term for having the computer perform the instructions of your program. Alias for “run it.”

extension, Jupyter: in this instance, it is not a request for more time. Rather, a Jupyter extension is a bit of code, often developed by a third-party, that adds additional functionality to Jupyter. For example, a popular extension is a Table of Contents creator.

flipped classroom: a teaching style where students work on their own outside of class to learn new material (sometimes by watching recorded lectures or reading descriptive/interactive notebooks) and then come together in the classroom to practice what they’ve learned through exercises or experiments.

Git: a popular version control system (VCS) used for keeping track of changes of files over time.

IDE (Integrated Development Environment): software that assists in the development of additional software.

Jupyter: The term “Jupyter” may refer to one of a couple of different things: a community of users and developers focused on the open source software; the collection of tools and standards that, together, allow projects like the Jupyter Notebook to operate. The name refers to the three core programming languages supported: Julia, Python, and R.

JupyterHub: a cloud service that can provide access to Jupyter notebooks and environments to multiple users via a modern web browser. <http://jupyter.org/hub>

kernel: In Jupyter, a kernel is the packaging up of a language, and related programs needed to run it. For example, Python2 and Python3 are separate kernels.

LMS (Learning Management System): a cloud service that helps instructors manage aspects of classrooms.

load: how many students can a computer support?

Markdown: a text format that allows for basic formatting (headers, text styles, links) mixed inline with the text. Markdown files usually have the extension `.md` and can be rendered natively by GitHub and other tools.

magic: a meta-command typically starting with one or two percent signs. Changes the meaning of the contents of a line (one percent sign, `%`) or the cell (two percent signs, `%%`) from code to a particular meta-instruction. For example, `%%R` indicates that the cell contents will be interpreted as commands to the R language. Magics are kernel-specific (e.g., vary with the kernel in use).

nbgrader: a tool for creating, handling, and automatically grading assignments based on Jupyter notebooks. <https://nbgrader.readthedocs.io>

nbviewer: a web application for rendering Jupyter notebooks as static web pages, providing a URL to share and view them with a modern web browser. <https://nbviewer.jupyter.org>

nbconvert: a tool for converting Jupyter notebooks into other formats such as PDF, HTML, LaTeX, Markdown, reStructuredText, and others. <https://nbconvert.readthedocs.io>

notebook hidden state: a technical term referring to the value of variables that may have surprising results due to cells having been executed in a non-sequential order.

open source: software and documents that are created in a manner that give you rights to be able to use, and reproduce.

pattern: A “pattern” is a technical term referring to an abstract description of a labeled process. For example, “wash, rinse, repeat” is a common pattern for cleaning various objects.

service, JupyterHub: JupyterHub can take advantage of additional separate, but integrated, software extensions. These are called “services.”

script: a colloquial term for a computer program.

unit test: a technical term for a “test” for checking to see if software is operating correctly.

URL (Universal Resource Locator): the address of a resource (e.g., webpage) on the internet.

widget: a user interface (such as buttons, sliders, and checkboxes) that allow the easy control of hidden computer code.