


Python Primer for Data Science

The Data Scientist

Dr. Yves J. Hilpisch 

March 27, 2026



**THE DATA
SCIENTIST**

¹Get in touch: <https://linktr.ee/dyjh>. Web page: <https://thedata scientist.dev>. Research, structuring, drafting, and visualizations were assisted by GPT 5.x as a co-writing tool under human direction. Comments and feedback are welcome.

Preface

This primer is the optional ramp-up for *The Data Scientist*. It exists for one practical reason: the first core module should be about Python itself, not about fighting with setup, folders, terminals, or uncertainty about where work belongs.

Many beginners arrive with a mixed kind of hesitation. They may be willing to learn Python, but the surrounding workflow still feels vague. Where should code run? What is a notebook actually for? What happens when packages are installed? How should work be saved so it is easy to revisit later? Those questions are not the center of the whole program, but they are large enough to block momentum if they remain unanswered.

That is why the primer exists. It is a runway, not the destination. It is meant to lower friction, establish a calm working rhythm, and help the reader enter Module 1 with the feeling that the surrounding tools are understandable enough to stop dominating attention.

Who This Primer Is For

The primer is especially useful for:

- absolute beginners,
- returners who have touched Python before but feel rusty,
- readers who are comfortable with ideas but not with tooling,
- and delegates who want a cleaner, more deliberate learning workflow from day one.

The intended style is deliberately practical. The primer does not try to teach every possible tool or workflow. Instead, it chooses a small set of habits that transfer well:

- notebook-first exploration,
- terminal literacy without unnecessary complexity,
- minimal Git discipline,
- local environments that are understandable rather than magical,
- and a portfolio mindset from the start.

One theme runs through the whole book: keep the basic workflow visible. A beginner often struggles less with individual concepts than with invisible workflow assumptions. Once those assumptions become explicit, progress becomes much steadier.

How to Read This Book

Move quickly, but do not move carelessly. Run the tiny examples. Save small artifacts. Write short notes. If something feels unclear, reduce the scope and rerun a smaller version. The purpose of the primer is not to create depth for its own sake; the purpose is to create readiness for the work that follows.

By the end of the primer, the reader should be able to:

- open and use a notebook responsibly,
- understand a minimal local Python workflow,
- keep files and environments conceptually separate,
- use Git for small checkpoints,
- and produce one tiny but complete portfolio artifact.

That is enough. The primer succeeds when it makes the first real module feel lighter, not when it becomes a world of its own.

How This Primer Fits into the Program

This book, *Python Primer for Data Science* [1], is the optional runway for the full five-book program. The core arc then continues with:

- *Python Programming Foundations for Data Science* [2], which turns basic workflow readiness into solid Python problem-solving,
- *Python Data Analysis, Visualization, and Storytelling* [3], which moves from Python code to arrays, tables, and charts,
- *Statistics, Machine Learning, and Model Evaluation* [4], which adds probabilistic thinking and baseline models, and
- *Software Engineering, Reproducibility, and Deployment Basics* [5], which focuses on projects, reliability, and packaging.

When a chapter in this primer mentions a later skill — plotting, table work, model evaluation, or packaging — it is pointing toward those later volumes. Reading the primer is not required, but finishing it should make the transition into Module 1 [2] and the rest of the series feel noticeably calmer.

Contents

Preface	i
1 Welcome and Study Logic	1
1.1 What This Program Is Actually Trying to Do	2
1.2 Why the Primer Exists	2
1.3 How the Books Fit Together	2
1.4 The Study Loop	3
1.5 What Self-Paced Actually Means	3
1.6 What Counts as Progress	3
1.7 Where We Are Heading Next	4
2 Colab and Notebook Basics	5
2.1 Why Notebooks Matter	6
2.2 The Two Main Kinds of Cells	6
2.2.1 Markdown Cells	6
2.2.2 Code Cells	6
2.3 Why Colab Is a Good First Home	7
2.4 A Minimal Notebook Pattern	7
2.5 Execution Order and State	8
2.6 Markdown Is Not Decoration	8
2.7 A Tiny First Example	8
2.8 When a Notebook Is Enough	8
2.9 Moving a Notebook from Colab to Your Local Project	9
2.9.1 The Colab Download Path	9
2.9.2 The Local Notebook Path	9
2.10 Where We Are Heading Next	10
3 Local Setup and Tools	11
3.1 Why Python Environments Exist	12
3.2 Why the .venv Convention Became Popular	12
3.3 Why an In-Project .venv Can Be a Problem	12
3.4 A Better Mental Model: Project Here, Environment There	13
3.5 A Good Alternative on macOS	13
3.6 Creating a Local Virtual Environment Step by Step	14
3.6.1 macOS and Linux	14
3.6.2 Windows	14
3.6.3 What Activation Actually Means	15
3.6.4 How to Check That It Worked	15
3.6.5 A Tiny Sanity Check	15
3.7 When a Local .venv Still Makes Sense	15
3.8 Use <code>python -m pip</code> , Not Bare <code>pip</code>	16
3.9 Terminal Editors: nano First, vim Next	16

3.10	Why These Tools, and Why Not the Bigger IDEs First?	16
3.10.1	nano	17
3.10.2	vim	17
3.10.3	What You Actually Need to Learn First	17
3.11	Installing nano and vim on Linux, macOS, and Windows	18
3.11.1	Linux	18
3.11.2	macOS	18
3.11.3	Windows	18
3.12	Folders, Navigation, and Current Working Directory	19
3.13	A Minimal Local Workflow	19
3.14	Where We Are Heading Next	19
4	Git, Portfolio, and Workflow Habits	21
4.1	Why Version Control Matters Even for Beginners	22
4.2	A Gentle Mental Model for Git	22
4.3	The Smallest Useful Git Workflow	22
4.4	A One-Time Git Setup Many Beginners Need	23
4.5	Why GitHub Enters the Picture	23
4.6	Portfolio Structure	23
4.7	The Role of README.md	24
4.8	Learning Journals and Reflection	24
4.9	Naming and Small Checkpoints	24
4.10	What Makes Work Portfolio-Ready	24
4.11	Where We Are Heading Next	25
5	First Mini Project	26
5.1	What the Mini Project Should Be	27
5.2	A Good Analogy: The First Finished Brick	27
5.3	A Concrete Shape for the Notebook	27
5.4	An Example Mini Project Idea	28
5.5	What to Save Alongside the Notebook	28
5.6	What Good Looks Like	28
5.7	Exit Criteria for the Primer	29
5.8	Where We Are Heading Next	29
	Glossary	31
	Epilogue	35

List of Figures

1.1	The study loop used throughout the program.	3
2.1	A notebook has two jobs: explain what is happening and execute the work.	6
3.1	Keep project files visible and keep large environments out of synced or tracked folders when possible.	13

Contact

Python Primer for Data Science

The Data Scientist



**THE DATA
SCIENTIST**

Get in touch:

linktr.ee/dyjh

thedata scientist.dev