



References: Writing Correct and Readable Code

Basics: An example workflow | *Snakemake 9.20.0 documentation*. (n.d.). Retrieved May 5, 2026, from <https://snakemake.readthedocs.io/en/stable/tutorial/basics.html>

black: The uncompromising code formatter. (Version 26.3.1). (n.d.). [Python; OS Independent].

Classic packaging. (n.d.). Scientific Python Development Guide. Retrieved May 5, 2026, from <https://learn.scientific-python.org/development/guides/packaging-classic/>

DANDI Archive. (n.d.). Retrieved May 5, 2026, from <https://dandiarchive.org/dandiset/000715/0.241009.1514>

Debugging code—Scientific Python Lectures. (n.d.). Retrieved May 5, 2026, from <https://lectures.scientific-python.org/advanced/debugging/index.html#using-the-python-debugger>

docopt—Language for description of command-line interfaces. (n.d.). Retrieved May 5, 2026, from <http://docopt.org/>

flake8: The modular source code checker: pep8 pyflakes and co (Version 7.3.0). (n.d.). [Python]. Retrieved May 5, 2026, from <https://github.com/pycqa/flake8>

Getting Started with GitHub—Pythia Foundations. (n.d.). Retrieved May 5, 2026, from <https://foundations.projectpythia.org/foundations/getting-started-github/>

Johnson, T. (2024, June 2). Useful tips for logging in Python. *Medium*. <https://medium.com/@tyrel.j.johnson/useful-tips-for-logging-in-python-c2c945358b5e>

Lab Teaching: Docker 101 for reproducible science. (n.d.). Kording Lab at the University of Pennsylvania. Retrieved May 5, 2026, from <https://kordinglab.com/2022/10/28/LabTeaching-Docker-for-Science.html>

Learn—Typer. (n.d.). Retrieved May 5, 2026, from <https://typer.tiangolo.com/tutorial/>

Master Python's argparse Module: Build Better CLIs. (n.d.). Retrieved May 5, 2026, from <https://www.datacamp.com/tutorial/python-argparse>

Mineault, P., & community, T. G. R. C. H. (2021). *The Good Research Code Handbook* [Computer software]. Zenodo. <https://zenodo.org/record/5796873>

Morgunov, A. (2026, January 25). *A Comprehensive Guide to Python Dependency Management with uv*. Anton Morgunov. <https://ischemist.com/writings/tutorials/comprehensive-guide-to-uv>

mypy: Optional static typing for Python (Version 1.20.2). (n.d.). [Python]. Retrieved May 5, 2026, from <https://www.mypy-lang.org/>

Object-oriented programming (OOP)—Scientific Python Lectures. (n.d.). Retrieved May 5, 2026, from <https://lectures.scientific-python.org/intro/language/oop.html>

PEP 8 – Style Guide for Python Code | peps.python.org. (n.d.). Python Enhancement Proposals (PEPs). Retrieved May 5, 2026, from <https://peps.python.org/pep-0008/>

Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do Users Write More Insecure Code with AI Assistants? *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS '23*, 2785–2799. <https://doi.org/10.1145/3576915.3623157>

pipreqs: Pip requirements.txt generator based on imports in project (Version 0.5.0). (n.d.). [Python]. Retrieved May 5, 2026, from <https://github.com/bndr/pipreqs>

Pryke, B. (2025, October 23). How to Use Jupyter Notebook: A Beginner's Tutorial. *Dataquest*. <https://www.dataquest.io/blog/jupyter-notebook-tutorial/>

pylint: Python code static checker (Version 4.0.5). (n.d.). [Python; OS Independent].

Pytest documentation. (n.d.). Retrieved May 5, 2026, from <https://docs.pytest.org/en/stable/>

Python Data Types. (n.d.). Retrieved May 5, 2026, from

https://www.w3schools.com/python/python_datatypes.asp

Python Poetry: Modern And Efficient Python Environment And Dependency Management.

(n.d.). Retrieved May 5, 2026, from <https://www.datacamp.com/tutorial/python-poetry>

Python, R. (n.d.-a). *Python IDEs and Code Editors (Guide) – Real Python*. Retrieved May 5,

2026, from <https://realpython.com/python-ides-code-editors-guide/>

Python, R. (n.d.-b). *Ruff: A Modern Python Linter for Error-Free and Maintainable Code –*

Real Python. Retrieved May 5, 2026, from <https://realpython.com/ruff-python/>

ruff: An extremely fast Python linter and code formatter, written in Rust. (Version 0.15.12).

(n.d.). [Python, Rust; OS Independent]. Retrieved May 5, 2026, from

<https://docs.astral.sh/ruff>

Saharkhiz, S., Chan, W., Kirezi, C. B., Petite, M., Roenspies, T., Perkins, T. J., & Colavita,

A. (2025). *VNC-Dist: A machine learning-based semi-automated pipeline for*

quantification of neuronal positioning in the C. elegans ventral nerve cord (p.

2024.11.16.623955). bioRxiv. <https://doi.org/10.1101/2024.11.16.623955>

Security Degradation in Iterative AI Code Generation: A Systematic Analysis of the

Paradox. (n.d.). Retrieved May 5, 2026, from

<https://arxiv.org/html/2506.11022v1#S2.T1>

Singh, K. (2021, March 13). Complete tutorial on how to use Hydra in Machine Learning

projects. *Towards Data Science*.

<https://towardsdatascience.com/complete-tutorial-on-how-to-use-hydra-in-machine-learning-projects-1c00efcc5b9b/>

Sphinx—Sphinx documentation. (n.d.). Retrieved May 5, 2026, from

<https://www.sphinx-doc.org/en/master/>

Testing with pytest. (n.d.). Scientific Python Development Guide. Retrieved May 5, 2026,

from <https://learn.scientific-python.org/development/guides/pytest/>

unittest—Unit testing framework. (n.d.). Python Documentation. Retrieved May 5, 2026,

from <https://docs.python.org/3/library/unittest.html>

Vatsal. (2022, May 9). Comprehensive Guide to Python Virtual Environments using Conda for Data Scientists. *Towards Data Science*.

<https://towardsdatascience.com/comprehensive-guide-to-python-virtual-environments-using-conda-for-data-scientists-6ebea645c5b/>

Version Control (Git). (n.d.). Missing Semester. Retrieved May 5, 2026, from

<https://missing.csail.mit.edu/2020/version-control/>

Writing documentation. (n.d.). Scientific Python Development Guide. Retrieved May 5,

2026, from <https://learn.scientific-python.org/development/guides/docs/>

Yemini, E., & Rusch, K. (2025). *QuPath Segmentation masks of NeuroPAL Images*

[Dataset]. Zenodo. <https://doi.org/10.5281/zenodo.14873352>