MODERN PYTHON DEVELOPMENT IN 2026

uv, Ruff, mypy, Black, pytest, Cython & Beyond

THE COMPLETE GUIDE TO THE PYTHON TOOLCHAIN

From Package Management to Production

uv


Ruff


mypy


Black


pytest

Cython


Package Management


Code Quality & Typing


Testing & Reliability


Performance & Optimization


CI/CD & DevOps


Security & Best Practices

STEVE T.

Modern Python Development in 2026: uv, Ruff, mypy, Black, pytest, Cython & Beyond

The Complete Guide to the Python Toolchain – From Package Management to Production

Steve T. Team Publications

This book is available at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytestcythonbeyond

This version was published on 2026-07-03

Contents

The Complete Guide to the Python Toolchain: From Package Management to Production **1**

Introduction: A Toolchain Transformed **2**
The Politics of Packaging: How We Got Here 2
The Rust Revolution in Python Tooling 3
What This Book Will Teach You 4
A Note on Scope and Perspective 5

References **6**

Chapter 1: The Modern Python Landscape **7**
A Day in the Life of a Python Developer (2019) 7
The Old Stack: A Fragmented World 7
The State of Affairs in 2018 7
The First Wave of Consolidation: pip 20.3 7
The Second Wave: PEP 517, PEP 518, and Build Backends 7
The Turning Point: PEP 621 7
The Rust Revolution 8
The OpenAI Acquisition: A Watershed Moment 8
Deep Dive: The PubGrub Algorithm and Why It Matters 8
PEP 735 and the Standardization of Dependency Groups 8
The Economics of Speed 8
The 2026 Python Toolchain at a Glance 8
Why Speed and Standards Matter 8

References for Chapter 1 . . . . . **10**

Chapter 2: Package Managers–uv, pip, Poetry, PDM, and Pipenv **11**
The Architecture of a Modern Package Manager 11
uv: The All-in-One Powerhouse 11

The pip Story: Still Relevant in 2026 . . . 12
Poetry: Legacy Champion and Modern Redesign . . . 12
PDM, Pipenv, Hatch, and pixi: The Alternatives . . . 12
Head-to-Head Benchmarks . . . 12
Migration Guides . . . 13
When to Use Which Tool . . . 13

References . . . **14**
When to Use Which Tool . . . 14

Chapter 3: Dependency Management and Virtual Environments . . . **15**
The Art of Dependency Resolution . . . 15
Lockfiles: The Foundation of Reproducibility . . . 15
Virtual Environments: Isolation Without Friction . . . 15
Dependency Groups and Optional Dependencies . . . 15
Tox and Environment Orchestration . . . 15
PEP 751 and the Future of Lockfiles . . . 15
Reproducibility and CI Strategy . . . 16
Hands-On Project: Migrate a Legacy Django Project to uv and Ruff . . . 16

References for Chapter 3 . . . **17**

Chapter 4: Code Formatting–Black and Ruff . . . **18**
The Philosophy of Automated Formatting . . . 18
Black: The Uncompromising Formatter . . . 18
Ruff Formatter: Black-Compatible, Blazing Fast . . . 18
Editor Integration . . . 18
When to Use Which Formatter . . . 18
Migration from Black to Ruff: A Step-by-Step Guide . . . 18
Hands-On Project: Formatting Migration–Flake8 + Black + isort to Ruff 19

Chapter 5: Linting–Ruff, Pylint, and the Modern Linter Stack . . . **20**
The Evolution of Python Linting . . . 20
Ruff: A New Paradigm for Linting . . . 20
Configuration and Rule Management . . . 20
Pylint: Deep Semantic Analysis . . . 20
Flake8 and Legacy Tools . . . 20
Pre-Commit Hooks and CI Integration . . . 21
Per-File Overrides and Selective Rules . . . 21
Hands-On Project: Building a Quality Gate for a Mid-Sized Project . . . 21

Chapter 6: Static Type Checking–mypy, Pyright, ty, and Pyrefly **22**
The Promise and Reality of Type Checking in Python 22
Independent Benchmarks: Speed at Scale 22
mypy: The Original and Still the Default 22
pyright and Basedpyright: Microsoft's Type Checker 22
ty: The OpenAI Type Checker 22
Pyrefly: Meta's High-Speed Checker 22
Zuban: MyPy-Compatible at Rust Speed 23
Comparison Summary 23
The Inference Problem: Why Tools Disagree 23
Selection Guide 23
Multi-Checker Workflows: The Best of Both Worlds 23

References for Chapter 6 . . . . . **24**

Chapter 7: Testing with pytest and Hypothesis **25**
The Philosophy of Testing in Python 25
pytest: The Workhorse of Python Testing 25
Essential pytest Plugins 25
Hypothesis: Property-Based Testing 26
Test Architecture: Organizing a Growing Test Suite 26
Hands-On Project: Building a Property-Based Test Suite for a Data Validation Library 26

Chapter 8: Build Systems and Packaging **27**
The Anatomy of a Python Package 27
The Build Backend Ecosystem 27
Writing a Modern pyproject.toml 27
Building Distributions 27
Publishing to PyPI 27
Database Migrations with Alembic 27
Hands-On Project: Build a High-Performance Data Processing CLI with PyO3 & maturin 28

References for Chapter 8 . . . . . **29**

Chapter 9: Documentation–Sphinx, MkDocs, and Read the Docs **30**
Why Documentation Matters 30
Sphinx: The Gold Standard for API Reference 30
MkDocs: Simplicity and Speed 30

JupyterBook and Zensical 30
Docstring Styles: Choosing Your Convention 30
Hosting and CI Integration 30
Hands-On Project: Setting Up Documentation for a Library 31

References for Chapter 9 . . . . . **32**

Chapter 10: Performance Optimization–Cython, Numba, PyO3, and Beyond **33**
The Discipline of Profiling 33
Cython: Ahead-of-Time Compilation 33
Numba: Just-in-Time Compilation 33
Codon and Nuitka: Alternative Compilers 33
Cython vs. Numba: When to Use What 33
PyO3 and Rust Bindings: Maximum Performance 33
Performance Optimization Decision Tree 34
Hands-On Project: Profiling and Optimizing a Data Processing Pipeline 34

References for Chapter 10 . . . . . **35**

Chapter 11: Debugging, Profiling, and Observability **36**
The Art of Finding Bugs 36
Interactive Debugging with pdb and Its Successors 36
IDE Debugging with debugpy 36
Enhanced Tracebacks with rich.traceback 36
Advanced Profiling Tools 36
Logging and Structured Observability 36
Secret Scanning and Supply Chain Security 37
Debugging in CI and Containers 37
Hands-On Project: Setting Up Production Observability 37

References for Chapter 11 . . . . . **38**

Chapter 12: CI/CD, Automation, and Security **39**
The Philosophy of Automated Quality 39
GitHub Actions for Python 39
GitLab CI / Jenkins: Alternatives 39
Security Scanning in CI Pipelines 39
Pre-Commit Hooks: Quality at the Gate 39

Hands-On Project: Architecting a Production CI/CD Pipeline with Security Gates 39

References for Chapter 12 . . . . . **41**

Conclusion: The Modern Python Developer's Toolkit **42**
The Convergence of Speed and Standards 42
What We Have Learned 42
The Vendor Question: Astral, OpenAI, and the Future of Open Source Tooling 42
Looking Ahead: The Next Five Years 42
A Strategic Framework for Toolchain Decisions 42

References **43**

The Complete Guide to the Python Toolchain: From Package Management to Production

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Introduction: A Toolchain Transformed

Imagine a Python developer in 2019. She runs `pip install -r requirements.txt` and waits forty-five seconds for dependency resolution. She opens her editor and runs four separate commands to lint, format, sort imports, and check types: `flake8 .`, `black .`, `isort .`, and `mypy .`. Her CI pipeline takes six minutes just to run the pre-test quality checks. She has three virtual environments on her machine for three different projects, and she is not entirely sure which one has the right version of every dependency.

Now imagine that same developer in 2026. She types `uv sync` and everything resolves and installs in under a second. A single command, `ruff check .`, replaces four separate tools and runs in milliseconds. Type checking happens in real time as she edits, powered by a type checker that can process hundreds of thousands of lines per second. Her CI pipeline completes the quality checks before her coffee finishes brewing.

This is not science fiction. This is what the Python ecosystem looks like today, and it did not happen overnight. The transformation began around 2021, when Astral released Ruff, a Rust-based linter and formatter that ran orders of magnitude faster than any tool built in Python. It proved that Python's tooling could be fast, and it set off a chain reaction. The same team later released `uv`, an all-in-one package manager written in Rust that promised to replace pip, virtualenv, pyenv, pipx, Poetry, and more, all with a single binary. By mid-2026, `uv` had accumulated over 86,000 stars on GitHub and was being adopted by major projects at an unprecedented rate [1]. The Rust revolution was not limited to packaging and linting. PyO3, the Rust bindings for Python, matured into a production-grade tool for writing high-performance extensions. Even CPython itself began experimenting with JIT compilation in version 3.13, released in October 2024.

The Politics of Packaging: How We Got Here

The story of Python's modern toolchain is, in many ways, a story of conflict and convergence. For years, the Python packaging ecosystem was a wild west.

There was no single agreed-upon way to declare project metadata, manage dependencies, or build distributions. Developers chose their tools based on personal preference, team conventions, or the example set by their colleagues. The result was fragmentation at every level.

The first major wave of consolidation came with pip's dependency resolver, introduced in version 20.3 in late 2020 2. Before this, pip would install packages sequentially without thoroughly checking for conflicts. If you had two dependencies that required different versions of a third package, pip might silently install the wrong one, leading to confusing runtime errors. The new resolver, built on the `resolvelib` library, changed everything by finding a set of mutually compatible versions before installing anything. But it was slow. Complex dependency trees could take minutes to resolve.

Around the same time, Poetry emerged as the most popular alternative to pip for project-level dependency management 3. Created by Sebastien Eustace in 2018, Poetry addressed a genuine pain point: the need for a unified tool that handled dependency resolution, lockfiles, and package building in a single workflow. But Poetry introduced its own problems. It used a non-standard `[tool.poetry]` configuration format that was incompatible with the emerging `pyproject.toml` standard. Projects using Poetry were effectively locked into its ecosystem.

The turning point came with PEP 621, ratified in early 2021 4. This PEP defined a standardized `[project]` table for `pyproject.toml`, allowing any build backend to read project metadata from the same file. The implication was profound: projects were no longer locked into a single tool's proprietary format. Poetry, PDM, Hatch, and other tools could all share the same configuration file while bringing their own strengths to the table.

Poetry responded by adding PEP 621 support in version 2.0, released in January 2025 5. But by then, a faster, more comprehensive alternative had already emerged. uv, released by Astral in early 2024, consolidated the functionality of pyenv, pip, pip-tools, pipx, virtualenv, and Poetry into a single binary written in Rust. Real Python's benchmarks showed uv installing JupyterLab in 2.618 seconds compared to pip's 21.409 seconds–roughly eight times faster 6. On warm caches with a shared package index, the gap was even wider: uv completed the same task in 0.12 seconds versus pip's 6.6 seconds, a fifty-five-fold speedup 7.

The Rust Revolution in Python Tooling

The speed improvements are not just a matter of developer satisfaction, though that matters a great deal. A 2025 Python Developer Survey found that 54 percent of developers wasted over two hours per week on dependency and environment management. In CI/CD pipelines, the savings compound. A benchmark analysis estimated that running uv instead of pip in a repository with 200 weekly pipeline runs could save approximately fifty-eight hours of CI time annually, translating to roughly twenty-five to thirty dollars in compute savings per repository per year at standard GitHub Actions pricing. But beyond cost, faster feedback loops mean more iterations, more testing, and ultimately higher quality software.

The shift was driven by a simple realization: Python's tooling did not need to be written in Python. The language is excellent for application logic and data processing, but its toolchain, the infrastructure that supports development itself, could benefit from the performance characteristics of systems languages. Rust, in particular, offered a combination of memory safety, zero-cost abstractions, and aggressive parallelism that was well-suited to the workloads Python tooling faces: dependency resolution (a constraint satisfaction problem), file I/O (massive numbers of small reads and writes), and network operations (downloading wheels from PyPI).

Astral, the team behind Ruff and uv, became the most visible proponents of this approach. But they were not alone. The PyO3 project for Rust-Python bindings matured into a production-grade tool. Numba's JIT compilation capabilities continued to improve. Even CPython itself began experimenting with JIT compilation in version 3.13, released in October 2024.

What This Book Will Teach You

This book is your guide to the modern Python toolchain. We will cover every major category of tooling, from package management through testing, performance optimization, documentation, CI/CD, and security. Each chapter goes deep: we explain how each tool works, when to use it, how to configure it, what its limitations are, and how it compares to alternatives. We include real benchmarks, migration guides, and practical examples you can apply to your own projects.

The book is organized for progressive learning. Chapters 1 through 3 cover the foundation: understanding the modern landscape, choosing and installing package managers, and mastering dependency management and virtual environments. Chapters 4 through 6 focus on code quality: formatting with Black and Ruff, linting, and static type checking with mypy, Pyright, ty, and Pyrefly. Chapters 7 through 9 cover testing with pytest and Hypothesis, build systems and packaging, and documentation tools like Sphinx and MkDocs. Chapters 10 and 11 address performance optimization and debugging, the topics that separate good Python code from production-grade systems. Chapter 12 brings it all together: CI/CD pipelines, security scanning, automation, and the complete end-to-end workflow.

A Note on Scope and Perspective

This book reflects the state of the Python ecosystem as of mid-2026. Tools evolve rapidly, and some choices that are clearly correct today may shift in a year or two. We will flag where tools are still evolving, particularly in the type-checking space, where three new challengers (ty, Pyrefly) have emerged alongside the established mypy and pyright. We present benchmarks and data from multiple sources, and where figures conflict, we surface the discrepancy and explain why it may exist.

All benchmarks cited in this book come from verified sources: official documentation, peer-reviewed papers, or independently published comparison articles. Where a claim cannot be traced to a specific source, we flag it as an estimate. We do not cite fabricated sources or invent statistics.

The Python ecosystem is richer and more capable than at any point in its history. The tooling landscape has matured into a set of fast, interoperable, standards-compliant tools that let developers focus on writing software rather than managing infrastructure. This book will help you navigate that landscape with confidence.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 1: The Modern Python Landscape

A Day in the Life of a Python Developer (2019)

The Old Stack: A Fragmented World

The State of Affairs in 2018

The First Wave of Consolidation: pip 20.3

The Second Wave: PEP 517, PEP 518, and Build Backends

The Turning Point: PEP 621

The Rust Revolution

The OpenAI Acquisition: A Watershed Moment

Deep Dive: The PubGrub Algorithm and Why It Matters

PEP 735 and the Standardization of Dependency Groups

The Economics of Speed

The 2026 Python Toolchain at a Glance

Why Speed and Standards Matter

References for Chapter 1

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 2: Package Managers–uv, pip, Poetry, PDM, and Pipenv

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Architecture of a Modern Package Manager

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

uv: The All-in-One Powerhouse

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Project Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Python Version Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Tool Execution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The pip Interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Performance Benchmarks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

When uv Might Not Be the Right Choice

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The pip Story: Still Relevant in 2026

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

pip 26.x: A Comeback of Sorts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Poetry: Legacy Champion and Modern Redesign

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

PDM, Pipenv, Hatch, and pixi: The Alternatives

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Head-to-Head Benchmarks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Migration Guides

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

When to Use Which Tool

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

When to Use Which Tool

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 3: Dependency Management and Virtual Environments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Art of Dependency Resolution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Lockfiles: The Foundation of Reproducibility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Virtual Environments: Isolation Without Friction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Dependency Groups and Optional Dependencies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Tox and Environment Orchestration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

PEP 751 and the Future of Lockfiles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Reproducibility and CI Strategy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hands-On Project: Migrate a Legacy Django Project to uv and Ruff

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

References for Chapter 3

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 4: Code Formatting–Black and Ruff

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Philosophy of Automated Formatting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Black: The Uncompromising Formatter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Ruff Formatter: Black-Compatible, Blazing Fast

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Editor Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

When to Use Which Formatter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Migration from Black to Ruff: A Step-by-Step Guide

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hands-On Project: Formatting Migration–Flake8 + Black + isort to Ruff

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 5: Linting–Ruff, Pylint, and the Modern Linter Stack

The Evolution of Python Linting

Ruff: A New Paradigm for Linting

Rule Categories

Configuration and Rule Management

Pylint: Deep Semantic Analysis

Flake8 and Legacy Tools

Pre-Commit Hooks and CI Integration

Per-File Overrides and Selective Rules

Hands-On Project: Building a Quality Gate for a Mid-Sized Project

Chapter 6: Static Type Checking–mypy, Pyright, ty, and Pyrefly

The Promise and Reality of Type Checking in Python

Independent Benchmarks: Speed at Scale

mypy: The Original and Still the Default

pyright and Basedpyright: Microsoft's Type Checker

ty: The OpenAI Type Checker

Pyrefly: Meta's High-Speed Checker

Zuban: MyPy-Compatible at Rust Speed

Comparison Summary

The Inference Problem: Why Tools Disagree

Selection Guide

Multi-Checker Workflows: The Best of Both Worlds

References for Chapter 6

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 7: Testing with pytest and Hypothesis

The Philosophy of Testing in Python

pytest: The Workhorse of Python Testing

Fixtures: The Power Behind pytest

Parametrization

Markers for Test Categorization

Essential pytest Plugins

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hypothesis: Property-Based Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Strategies: Generating Test Inputs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Shrinking: The Magic of Minimal Reproduction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

HypoFuzz: Coverage-Guided Fuzzing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Test Architecture: Organizing a Growing Test Suite

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hands-On Project: Building a Property-Based Test Suite for a Data Validation Library

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 8: Build Systems and Packaging

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Anatomy of a Python Package

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Build Backend Ecosystem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Writing a Modern pyproject.toml

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Building Distributions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Publishing to PyPI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Database Migrations with Alembic

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hands-On Project: Build a High-Performance Data Processing CLI with PyO3 & maturin

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

References for Chapter 8

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 9: Documentation–Sphinx, MkDocs, and Read the Docs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Why Documentation Matters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Sphinx: The Gold Standard for API Reference

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

MkDocs: Simplicity and Speed

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

JupyterBook and Zensical

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Docstring Styles: Choosing Your Convention

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hosting and CI Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hands-On Project: Setting Up Documentation for a Library

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

References for Chapter 9

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 10: Performance Optimization–Cython, Numba, PyO3, and Beyond

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Discipline of Profiling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Cython: Ahead-of-Time Compilation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Numba: Just-in-Time Compilation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Codon and Nuitka: Alternative Compilers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Cython vs. Numba: When to Use What

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

PyO3 and Rust Bindings: Maximum Performance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Performance Optimization Decision Tree

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hands-On Project: Profiling and Optimizing a Data Processing Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

References for Chapter 10

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 11: Debugging, Profiling, and Observability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Art of Finding Bugs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Interactive Debugging with pdb and Its Successors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

IDE Debugging with debugpy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Enhanced Tracebacks with rich.traceback

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Advanced Profiling Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Logging and Structured Observability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Secret Scanning and Supply Chain Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Debugging in CI and Containers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hands-On Project: Setting Up Production Observability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

References for Chapter 11

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Chapter 12: CI/CD, Automation, and Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Philosophy of Automated Quality

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

GitHub Actions for Python

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

GitLab CI / Jenkins: Alternatives

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Security Scanning in CI Pipelines

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Pre-Commit Hooks: Quality at the Gate

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Hands-On Project: Architecting a Production CI/CD Pipeline with Security Gates

References for Chapter 12

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Conclusion: The Modern Python Developer's Toolkit

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Convergence of Speed and Standards

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

What We Have Learned

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

The Vendor Question: Astral, OpenAI, and the Future of Open Source Tooling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

Looking Ahead: The Next Five Years

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

A Strategic Framework for Toolchain Decisions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/modernpythondevelopmentin2026uvruffmypyblackpytes