

Preface

This book systematically covers how to build, operate, and secure automated news and market monitoring with Python - from data acquisition through filtering and AI-assisted classification to reliable notifications and sustained production operation. A real-world use case serves as the running reference project throughout: a watch system that monitors financial news and reports the moment a specific company shows signs of going public.

What makes this book different: the concepts aren't just named, they're worked through on a complete, runnable system and, at the same time, placed in their broader context. For every central design question - which architecture, which storage, which classification approach, which notification channel, which automation - alternatives are named and weighed against each other, not just the solution chosen for the project.

What this book is

A systematically structured technical book on the architecture and operation of monitoring and alerting systems in Python. It covers Python, web APIs, data storage, LLM-assisted classification, notification architectures, automation, testing, and operations - each with decision criteria: why this approach, what alternatives exist, and when they tend to pay off instead. Every chapter connects the abstract question to its concrete implementation in the reference project.

What this book is not

It does not replace the reference documentation of individual libraries, nor a detailed introduction to database theory - for that, the relevant specialist documentation is referenced. It is, however, meant as a coherent reference work on the subject of watch systems: if you only need a single chapter later on (say, to choose a storage solution or to automate something), you should be able to read it in isolation too.

Who is this book for?

Basic computer literacy is assumed: creating files, installing programs, typing commands in a terminal. Prior experience with a programming language is helpful but not required - every project-specific concept is introduced along the way. The book is aimed both at beginners who want to build a complete system step by step, and at practitioners who want to look up individual design decisions (architecture, storage, automation).

How is this book structured?

Part A (Chapters 1-6) lays the foundation: tools, Linux basics, architecture, configuration, security, and your first own watcher. Part B (Chapters 7-13) deepens the reference project piece by piece: data acquisition, persistence, LLM classification, filter pipelines, notification channels, and automation. Part C (Chapters 14-15) covers testing as well as the transition from prototype to a production-ready, operated system - topics that are usually missing from pure beginner tutorials but belong in a proper technical book. Part D (Appendix) bundles a glossary, sample solutions, a troubleshooting checklist, and further resources.

Every chapter follows the same pattern: learning objective, context, concept, implementation, deep dive, exercise, summary. The deep-dive sections are deliberately written to go beyond the reference project and to name alternatives and trade-offs for professional use.

The case study behind the project

In the summer of 2026 I wanted to invest in EngineAI Robotics - a Chinese humanoid-robot company that was about to go public. The problem: there was no official ISIN yet, no trading on a Western exchange, and the news was scattered. The idea: an agent of my own that watches the key sources and reports immediately once EngineAI shares actually become purchasable.

The agent isn't spectacular: no neural network, no magic. But it works, has been running reliably for weeks, and took a few hours of development and a few hundred lines of code. That's the kind of tool that makes

CHAPTER 1

Installing and Setting Up Python

Tools, virtual environments, first scripts

• Learning objectives

After this chapter you'll have a working Python installation, be able to run scripts, understand virtual environments (venv), and be able to install packages with pip.

Chapter 1: Python basics

1.1 Aside: Windows users - setting up WSL2

This book works with Linux terminal commands, Cron, and systemd throughout. If you're on Windows, the easiest path isn't to translate every single command, but to run a real Linux environment directly under Windows: the Windows Subsystem for Linux (WSL2). This isn't an emulator - it's a full-fledged Linux kernel running alongside Windows, so every command in this book will work exactly as written.

Here's how to set it up:

Open PowerShell as administrator and run `wsl --install`. This installs WSL2 with Ubuntu as the default distribution.

Restart the machine if prompted.

The first time you launch Ubuntu (Start menu -> "Ubuntu"), you'll create a username and password - this becomes your Linux login from now on, independent of your Windows login.

From here on you're in a real Ubuntu environment. Every terminal command from Chapter 2 works identically to a native Linux machine.