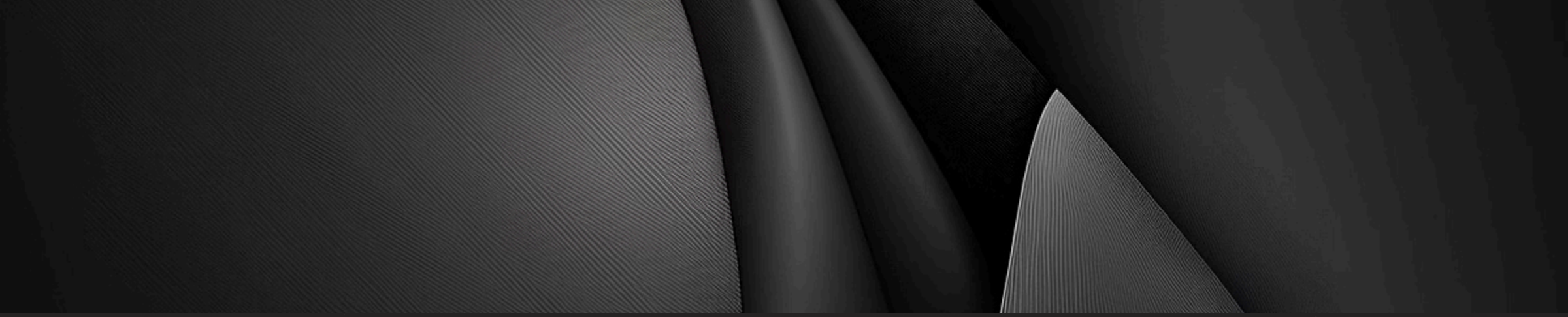

# Python Crash Course 2026

Covering Python 3.15 - All new improved version of Python

# Module 1: The Evolution & The Engine

Understanding the "Why" and the "How" of the interpreter.

**Key Topics:**

- History: From 1991 to the 3.15 release.
- The CPython VM: Bytecode compilation and the .pyc file.
- The 3.15 JIT: "Copy-and-Patch" execution and Tier 2 optimization.

# Module 2: Data Types & Structure

Mastering the high-level tools of the language.

**Key Topics:**

- The "Big Six": Integers, Strings, Dictionaries, Lists, Sets, and Tuples
- Logical Python: `If`, `For`, `While`, `Switch Case`
- The `collections` module: Specialized containers (Deque, Counter, NamedTuple).
- Pydantic & Typing: Enforcing data integrity at scale.

# Module 3: Python Libraries

The `os` and `sys` modules: Low-level system and environment interaction.

# Module 4: File Systems & Module Architecture

Mastering code organization and system I/O.

**Key Topics:**

- The Python File: Scripting vs. Module execution.
- Project Structure: Building robust packages with `pyproject.toml`.

# Module 5: Memory, GIL, & Internal Performance

Understanding how Python talks to the hardware.

**Key Topics:**

- Memory Allocation: Stack vs. Heap and the **slots** optimization.
- The Global Interpreter Lock (GIL): Why it exists and the new "No-GIL" builds.
- Sequence Memory: List over-allocation vs. Tuple immutability.
- Garbage Collection: Reference counting and generational cycles.

# Module 6: Advanced Python Concepts

Maximizing CPU and I/O performance.

**Key Topics:**

- Multi-processing: Bypassing the GIL for CPU-intensive tasks.
- Multi-threading: Handling I/O-bound operations effectively.
- AsyncIO: Mastering the Event Loop for high-concurrency applications.
- Context Manager

# Module 7: Modern Tooling and Packaging

Building a professional development environment.

**Key Topics:**

- The Evolution: `pip` (standard) vs. `conda` (data science).
- The Future: `uv`—the ultra-fast Rust-based package manager.
- Virtual Environments: Isolation strategies for system-level stability.

# Let's Get Started...