

# Mojo Manual

Welcome to the Mojo Manual, a complete guide to the Mojo  programming language!

Mojo is designed to solve a variety of AI development challenges that no other language can, because Mojo is the first programming language built from the ground-up with [MLIR](#) (a compiler infrastructure that's ideal for heterogeneous hardware, from CPUs and GPUs, to various AI ASICs). We also designed Mojo as a superset of Python because we love Python and its community, but we couldn't realistically enhance Python to do all the things we wanted. For a longer discussion on this topic, read [Why Mojo](#).

Beware that Mojo is still a very young language, so there's a lot that hasn't been built yet. Likewise, there's a lot of documentation that hasn't been written yet. But we're excited to share Mojo with you and [get your feedback](#).

## Contents

- **Get started**
  - [Why Mojo](#)
  - [Get started with Mojo](#)
- **Language basics**
  - [Introduction to Mojo](#)
  - [Functions](#)
  - [Variables](#)
  - [Types](#)
  - [Control flow](#)
  - [Structs](#)
  - [Modules and packages](#)
- **Value ownership**
  - [Intro to value ownership](#)
  - [Value semantics](#)
  - [Ownership and borrowing](#)
- **Value lifecycle**
  - [Intro to value lifecycle](#)
  - [Life of a value](#)

- [Death of a value](#)
- Traits and parameters
  - [Traits](#)
  - [Parameterization: compile-time metaprogramming](#)
- Pointers
  - [Unsafe pointers](#)
- Python
  - [Python integration](#)
  - [Python types](#)
- Tools
  - [Debugging](#)
  - [Testing](#)
- Project information
  - [Roadmap and sharp edges](#)
  - [Changelog](#)
  - [FAQ](#)

Was this page helpful?



Edit this page