

# 曾经号称比Python快9万倍，Mojo如今彻底开源了！

CSDN 2026年8月20日 14:41 江苏

CSDN

CSDN

关注技术世界的前沿动态，深入挖掘其背后的技术趋势与实践方法，助力技术人把握时代...

3334篇原创内容

公众号

整理 | 屠敏

出品 | CSDN (ID: CSDNnews)

曾经因为一句“比 Python 快 9 万倍”火遍开发者社区的 Mojo 语言，如今终于走到了一个真正值得关注的节点。

8 月 18 日，Swift 之父、编译器大神 Chris Lattner 和 Google 前员工 Tim Davis 共同创立的 Modular 公司正式宣布，**Mojo 整门语言正式全面开源**。

这一次不再只是开放标准库或者部分代码，而是连此前一直没有公开的 **Mojo 编译器、工具链以及构建语言所需的完整代码**，全部以 Apache 2.0（附 LLVM 例外条款）许可证放出。

一切都可以在 Modular GitHub 主仓库中获取：<https://github.com/modular/modular>



而就在一周前，Mojo 刚刚发布 1.0 正式版，开始进入源代码稳定阶段。也就是说，困扰 Mojo 社区多年的两个问题——“什么时候成熟？”以及“什么时候真正开源？”，几乎在同一时间有了答案。

更值得注意的是，就在今年 7 月，Qualcomm 已经完成了对 Modular 的收购。Modular 当时明确表示，Mojo、MAX 和 Modular Cloud 都将继续作为产品和品牌发展，同时坚持开放、异构的生态方向。

# // 01

## 这一次，Mojo 编译器也开源了

Mojo 从一开始就不是一门普通的编程语言。

它的目标，是让开发者能够使用一套语言和编译器，同时面向 CPU、GPU、AI 加速器等不同硬件进行高性能编程。Modular 希望借助现代编译器和编程语言技术，把过去需要大量底层代码和硬件适配工作的事情变得更加简单。

但过去四年里，Mojo 一直处于一种比较特殊的状态：社区是开放的，语言开发也是公开进行的，但编译器本身并没有开源。

这也是为什么这次开源的意义远不只是“又多了一个 GitHub 仓库”。

事实上，Mojo 的开源已经持续了很长时间。

2024 年 3 月，Mojo 标准库核心模块开始以 Apache 2.0 许可证开源，并开放社区贡献。此后，Modular 又陆续开放了大量 Mojo kernel、工具以及其他代码。

到了 2025 年，Modular 进一步简化 MAX 和 Mojo 的许可证，希望降低开发者使用 Mojo 的门槛。随后不久，Modular 还陆续开源了超过 45 万行生产级 Mojo kernel 和 serving 代码，让开发者能够直接看到 Mojo 如何用于 CPU、GPU 等高性能计算场景。

到了今年，Mojo 终于进入最后阶段。

5 月，Mojo 1.0 进入 Beta；8 月 11 日，Mojo 1.0 正式发布，Modular 希望以此建立一个更加稳定的语言版本。对于已经稳定的 API，1.x 系列将更加注重兼容性，不再像早期那样频繁改变语言本身。

紧接着，8 月 18 日，编译器和工具链正式开源。

Modular 表示，目前 Mojo 的全部代码都已经进入主 modular GitHub 仓库。开发者不仅可以阅读源码，还可以直接在本地构建 Mojo 编译器。

许可证方面，Mojo 采用 Apache 2.0，并加入 LLVM 例外条款。Modular 认为，这种组合能够给语言和编译器项目提供较大的使用自由度，也方便开发者构建和分发由 Mojo 编译出来的二进制文件。

换句话说，这次真正补上的，是 Mojo 此前一直缺失的最后一块拼图：编译器。

## // 02

## Mojo 早已不只是“Python++”

其实最早让众人了解 Mojo 的原因，是因为它看起来很像 Python。

2023 年首次公开时，Modular 曾把 Mojo 称为“Python++”：保留接近 Python 的语法和开发体验，同时加入静态类型、内存管理、SIMD、编译期元编程等能力，让开发者能够进一步向底层硬件和性能优化靠近。

随后，官方晒出“Mojo 比 Python 快 9 万倍”的说法又让它彻底出圈。

```
$ ls
LICENSE          deviceinfo.mojo  mandelbrot.mojo  notebooks
README.md        docker          matmul.mojo      pymatmul.py
__pycache__      hello.🔥        memset.mojo      reduce.mojo
blogs-videos     hello_interop.mojo nbody.mojo       simple_interop.py
$ mojo matmul.mojo
0.006192281958602655 GFL0P/s
Python           0.006 GFL0PS
Naive:           6.343 GFL0PS      1024.31x faster than Python
Vectorized:      22.245 GFL0PS     3592.45x faster than Python
Parallelized:    48.584 GFL0PS     7845.89x faster than Python
Tiled:           61.394 GFL0PS     9914.52x faster than Python
Unrolled:        64.004 GFL0PS     10336.11x faster than Python
Accumulated:     560.092 GFL0PS    90449.98x faster than Python
$
```

不过，这个数字来自特定的性能测试，并不能理解成“Mojo 在所有任务上都比 Python 快 9 万倍”。

更重要的是，Python 在 AI 和科学计算领域本身就不是简单靠解释器运行到底。大量真正消耗计算资源的代码，本来就由 NumPy、PyTorch 等底层 C/C++、CUDA 等实现完成。

所以，Mojo 真正想解决的问题，从来不是简单地“让 Python 变快”。

经过四年发展，Mojo 的定位也已经发生了变化。

它如今更多被放在 Modular 的 MAX AI 平台中，用来编写高性能 kernel，并连接 CPU、GPU 以及其他 AI 加速器。

Modular 现在对 Mojo 的定位，也已经从早期强调 Python 兼容性，逐渐转向面向 AI 时代的系统级编程语言。

这意味着，今天再看 Mojo，真正值得关注的问题已经不是“它能不能成为更快的 Python”，而是：能不能用一门语言，把 AI 应用从上层代码一路连接到底层硬件，同时尽可能减少针对不同芯片重复改写代码的工作。

这也是 Mojo 和 Python 的关系越来越值得重新理解的原因。

Python 生态依然可以存在，开发者也可以继续调用 Python 代码和库；但如果想真正利用 Mojo 的性能优势，往往还是需要使用 Mojo 自身的类型、内存和编译能力重新实现性能关键部分。

也就是说，Mojo 并不会让现有 Python 程序仅仅因为换了一个文件后缀，就突然快上几万倍。

## // 03

### 编译器开源了，但社区贡献还要再等等

不过，全面开源并不意味着 Mojo 从今天开始就完全交给社区。这恰恰是此次公告中另一个值得注意的地方。

事实上，Mojo 标准库从 2024 年开始就已经接受社区贡献，并在过去两年积累了大量来自开发者社区的反馈和代码。

但对于编译器和工具链，Modular 目前仍然采取相对谨慎的态度。

Modular 表示，**尤其是在 AI 编程快速发展的背景下，如何处理社区贡献需要更加谨慎。因此，现阶段还不会接受针对 Mojo 编译器和工具链的外部贡献。**

按照目前的计划，Modular 希望在 2026 年底之前开放编译器和工具链的社区贡献机制，届时将公布更多具体细节。

换句话说，这次开源已经允许开发者查看、构建和研究 Mojo 编译器源码，但如果想直接向上游提交编译器和工具链代码，目前还需要再等一段时间。

对于已经开始研究 Mojo 的开发者，Modular 目前则鼓励通过官方论坛讨论编译器源码、分享项目以及交流使用经验。

## // 04

### 现在就可以自己编译 Mojo

随着编译器开源，开发者现在可以直接从 Modular 的主 GitHub 仓库获取 Mojo 的完整源码。

对此，官方也直接给出了上手的教程。首先将代码仓库克隆到本地：

```
1 git clone https://github.com/modular/modular.git
2 cd modular
```

随后，可以通过一条 Bazel 命令从源码构建 Mojo 编译器，并直接运行 Mojo 文件：

```
1 ./bazelw run --config=build-mojo KGEN:mojo -- run hello.mojo
```

Modular 内部使用 Bazel 管理 Mojo 和 MAX 的复杂构建流程及缓存。上述 `--config=build-mojo` 参数会要求构建系统从头开始编译相关组件，并使用本地系统中的源代码。

开发者也可以直接修改 Mojo 编译器或标准库代码，然后运行完整的测试套件：

```
1 ./bazelw test --config=build-mojo mojo/stdlib/test/...
```

如果并不需要参与编译器本身的开发，则可以使用 `--config=prebuilt-mojo` 参数，由构建系统直接下载最新的 nightly 版本编译器二进制文件，从而减少本地编译所需的时间。

不过，目前如果需要定制 MAX kernel 或模型，仍然需要使用预构建的 Mojo 编译器。

## // 05

### 这一次，开发者的态度明显变了

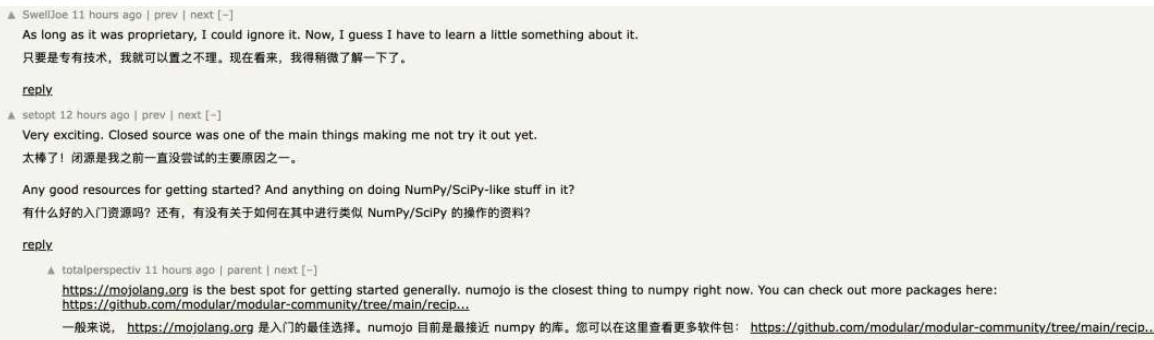
如果把时间拉回 Mojo 刚发布的时候，它在开发者社区的评价其实相当两极。

尤其是“比 Python 快 9 万倍”的说法，当时引发了大量 benchmark 争议。有人质疑测试条件，也有人认为直接拿 Mojo 和纯 Python 比较，很容易让读者对实际性能产生误解。

但四年过去，讨论的重点已经发生了变化。

这一次，Hacker News 上不少开发者讨论的已经不再是“9 万倍到底是不是真的”，而是 Mojo 现在到底能不能真正用起来。

有人直接表示，过去一直没有尝试 Mojo，其中一个重要原因就是编译器没有开源；现在源码终于公开，自己终于愿意认真试一试。



社区也开始讨论 Mojo 能否进一步补齐自己的生态。例如，已经有 NuMojo 等项目尝试提供类似 NumPy 的数值计算能力，并进一步探索 GPU 执行。

当然，质疑并没有消失。

有人认为 Windows 支持仍然不足，会影响 Mojo 的普及。也有人指出，Julia 已经在高性能计算、科学计算和数据科学领域发展多年，Mojo 并不是没有竞争对手。



这其实也是 Mojo 接下来真正需要面对的问题：

开源解决的是“为什么我不敢用”，但并没有自动回答“为什么我一定要用 Mojo”。

而这，可能比把编译器放到 GitHub 上更加重要。

## // 06

### 四年之后，Mojo 终于到了真正的分水岭

从 2023 年公开，到 2024 年开放标准库，再到 2025 年开放大量 kernel 和工具，最后到如今把编译器和工具链全部交出来，Mojo 花了四年时间完成了一次关键转变。

它已经不再只是一个“看起来很像 Python、但跑得更快”的实验性语言。

如今的 Mojo，正在尝试成为 Modular AI 软件栈中连接高层 AI 应用和底层异构硬件的一层基础设施。

而全面开源之后，Mojo 也终于要面对一个更加现实的问题：当编译器已经摆到开发者面前，当源码不再是门槛，它究竟能不能凭借自己的性能、硬件适配能力和开发体验，真正吸引开发者建立生态？

这一次，答案已经不再掌握在 Modular 一家公司手里。

代码已经开源，接下来真正决定 Mojo 能走多远的，可能就是开发者会拿它做什么。

参考：

<https://www.modular.com/blog/mojo-open-source>

<https://news.ycombinator.com/item?id=49348079>



云端筑基：重构下一代AI数据中心底座

GitHub大规模宕机7小时！马斯克旗下Cursor趁机“杀入”代码托管发布Origin，内部员工调侃：本想更早发布的

“我不会写代码，只是个Vibe Coder啊！”靠AI捣鼓的App爆火后，1亿美元大公司找他接单，他却慌了.....

🔊最后，说一件事

**2026 奇点智能技术大会与C++及系统软件技术大会终于要和大家见面了。**

11月20-21日·北京，奇点智能研究院联合CSDN，把两场技术大会放在了同一个时空里：

- 奇点智能技术大会（始于2016）——聊大模型、AI Native、企业级AI落地、多模态与世界模型；
- C++及系统软件技术大会（始于2005）——聊现代C++演进、AI算力与推理优化、高性能低时延系统。

为什么要放在一起？因为我们越来越相信——上层AI应用的爆发，离不开底层系统软件的支撑；而底层技术的演进方向，也正在被AI重新定义。

这次大会汇聚70+位技术专家、18个主题、1000+同行到场。

如果你也在这些方向上做研究、做产品、做工程，别错过。

奇点智能研究院  
Singularity Intelligence Research Institute

CSDN

奇点智能大会  
Singularity Intelligence Summit

## 奇点智能技术大会

Singularity Intelligence Technology Summit

## C++及系统软件技术大会

C++ and System Software Summit



11月20-21日



中国·北京万达文华酒店

70+

AI & 系统软件  
技术嘉宾

18大

前沿专题

1000+

行业精英

双会并行

### 演讲嘉宾

GUEST SPEAKER



Łukasz Kaiser

OpenAI研究科学家，  
GPT-5、GPT-4和  
Transformer共同创始人



李建忠

奇点智能研究院院长  
CSDN高级副总裁



张俊林

新浪微博首席科学家及  
AI研发部负责人



段楠

京东集团副总裁  
探索研究院副院长



方斌

北京邮电大学  
“拔尖人才”教授



张晨

小米AI平台部  
语言模型推理负责人



杨文婷

阿里云产品专家

更多嘉宾，敬请期待！



大会合作咨询



扫码领取大会资料

# SINGULARITY

