

Introduction 介绍

Getting started 开始

Installation 安装

First steps 第一步

Features 特征

Getting help 获取帮助

Guides 指南

Installing Python 安装 Python

Running scripts 运行脚本

Using tools 使用工具

Working on projects 从事项目

Publishing packages 发布包

Migration 迁移

Integrations 集成

Concepts 概念

Projects 项目

Tools 工具

Python versions Python 版本

Configuration files 配置文件

Package indexes 包索引

Resolution 分辨率

Build backend 构建后端

Authentication 认证

Caching 缓存

Preview features 预览功能

The pip interface pip 接口

Reference 参考

Getting started 开始

Installing uv 安装 uv

Installation methods 安装方法

Install uv with our standalone installers or your package manager of choice.
使用我们的独立安装程序或您选择的包管理器安装 uv。

Standalone installer 独立安装程序

uv provides a standalone installer to download and install uv:
UV 提供了一个独立的安装程序来下载和安装 UV:

macOS and Linux macOS 和 Linux Windows 窗户

```
$ curl -LsSf https://astral.sh/uv/install.sh | sh
```

Use `curl` to download the script and execute it with `sh`:
使用 `curl` 下载脚本并使用 `sh` 执行:

```
$ curl -LsSf https://astral.sh/uv/install.sh | sh
```

If your system doesn't have `curl`, you can use `wget`:
如果你的系统没有 `curl`, 你可以使用 `wget`:

```
$ wget -qO- https://astral.sh/uv/install.sh | sh
```

Request a specific version by including it in the URL:
通过在 URL 中包含特定版本来请求特定版本:

```
$ curl -LsSf https://astral.sh/uv/0.8.4/install.sh | sh
```

💡 Tip 提示

The installation script may be inspected before use:
使用前可以检查安装脚本:

macOS and Linux macOS 和 Linux Windows 窗户

```
$ curl -LsSf https://astral.sh/uv/install.sh | less
```

Alternatively, the installer or binaries can be downloaded directly from [GitHub](#).
或者, 可以直接从 [GitHub](#) 下载安装程序或二进制文件。

See the reference documentation on the [installer](#) for details on customizing your uv installation.
有关自定义 uv 安装的详细信息, 请参阅[安装程序](#)上的参考文档。

PyPI PyPI 的

For convenience, uv is published to [PyPI](#).
为方便起见, uv 发布到 [PyPI](#)。

If installing from PyPI, we recommend installing uv into an isolated environment, e.g., with `pipx`.
从 PyPI 安装, 我们建议将 uv 安装到隔离环境中, 例如使用 `pipx`:

```
$ pipx install uv
```

However, `pip` can also be used:
但是, `pip` 也可以用于:

```
$ pip install uv
```

📌 Note 注意

uv ships with prebuilt distributions (wheels) for many platforms; if a wheel is not available for a given platform, uv will be built from source, which requires a Rust toolchain. See the [contributing setup guide](#) for details on building uv from source.
UV 附带了适用于许多平台的预构建发行版 (轮子) ;如果给定的轮子不可用 平台, uv 将从源代码构建, 这需要一个 Rust 工具链。请参阅 [贡献设置指南](#) 有关从源代码构建 UV 的详细信息。

Homebrew 自制

uv is available in the core Homebrew packages.
uv 在核心 Homebrew 包中可用。

```
$ brew install uv
```

WinGet WinGet (温格)

uv is available via [WinGet](#).
uv 可通过 [WinGet](#) 获得。

```
$ winget install --id=astral-sh.uv -e
```

Scoop 舀

uv is available via [Scoop](#).
uv 可通过 [Scoop](#) 获得。

```
$ scoop install main/uv
```

Docker 码头工人

uv provides a Docker image at [ghcr.io/astral-sh/uv](#).
uv 在 [ghcr.io/astral-sh/uv](#)。

See our guide on [using uv in Docker](#) for more details.
有关更多详细信息, 请参阅我们在 [Docker 中使用 uv](#) 的指南。

GitHub Releases GitHub 发布

uv release artifacts can be downloaded directly from [GitHub Releases](#).
UV 释放伪影可直接从 [GitHub 发布](#)。

Each release page includes binaries for all supported platforms as well as instructions for using the standalone installer via `github.com` instead of `astral.sh`.
每个发布页面都包含所有受支持平台的二进制文件, 以及通过 `github.com` 而不是 `astral.sh` 使用独立安装程序的说明。

Cargo 负责

uv is available via Cargo, but must be built from Git rather than [crates.io](#) due to its dependency on unpublished crates.
uv 可通过 Cargo 获得, 但由于它依赖于未发布的 crate, 因此必须从 Git 而不是 [crates.io](#) 构建。

```
$ cargo install --git https://github.com/astral-sh/uv uv
```

📌 Note 注意

This method builds uv from source, which requires a compatible Rust toolchain.
此方法从源代码构建 uv, 这需要兼容的 Rust 工具链。

Upgrading uv 升级紫外线

When uv is installed via the standalone installer, it can update itself on-demand:
当 uv 通过独立安装程序安装时, 它可以按需更新:

```
$ uv self update
```

💡 Tip 提示

Updating uv will re-run the installer and can modify your shell profiles. To disable this behavior, set `UV_NO_MODIFY_PATH=1`.
更新 uv 将重新运行安装程序, 并可以修改您的 shell 配置文件。若要禁用此行为, 请设置 `UV_NO_MODIFY_PATH=1`。

When another installation method is used, self-updates are disabled. Use the package manager's upgrade method instead. For example, with `pip`:

```
$ pip install --upgrade uv
```

Shell autocompletion Shell 自动补全

💡 Tip 提示

You can run `echo $SHELL` to help you determine your shell.
您可以运行 `echo $SHELL` 来帮助确定 shell。

To enable shell autocompletion for uv commands, run one of the following:
要为 uv 命令启用 shell 自动完成, 请运行以下命令之一:

Bash 猛击 Zsh fish 鱼 Elvish 精灵 PowerShell / pwsh

```
echo 'uv generate-shell-completion fish | source' > ~/.config/fish/completions/uv.
```

To enable shell autocompletion for uvx, run one of the following:
要为 uvx 启用 shell 自动完成功能, 请运行以下作之一:

Bash 猛击 Zsh fish 鱼 Elvish 精灵 PowerShell / pwsh

```
echo 'uvx --generate-shell-completion fish | source' > ~/.config/fish/completions/uvx.
```

Then restart the shell or source the shell config file.
然后重新启动 shell 或获取 shell 配置文件。

Uninstallation 卸载

If you need to remove uv from your system, follow these steps:
如果您需要从系统中去除紫外线, 请按照下列步骤作:

1. Clean up stored data (optional):
清理存储的数据 (可选):

```
$ uv cache clean  
$ rm -r "$(uv python dir)"  
$ rm -r "$(uv tool dir)"
```

💡 Tip 提示

Before removing the binaries, you may want to remove any data that uv has stored.
在删除二进制文件之前, 您可能希望删除 uv 存储的任何数据。

2. Remove the uv and uvx binaries:
删除 uv 和 uvx 二进制文件:

macOS and Linux macOS 和 Linux Windows 窗户

```
$ rm ~/.local/bin/uv ~/.local/bin/uvx
```

📌 Note 注意

Prior to 0.5.0, uv was installed into `~/.cargo/bin`. The binaries can be removed from there to uninstall. Upgrading from an older version will not automatically remove the binaries from `~/.cargo/bin`.
在 0.5.0 之前, uv 被安装到 `~/.cargo/bin` 中。二进制文件可以从那里删除, 以卸载。从旧版本升级不会自动从 `~/.cargo/bin` 中。

Next steps 后续步骤

See the [first steps](#) or jump straight to the [guides](#) to start using uv.
请参阅[第一步](#)或直接跳转到指南以开始使用 uv。

Installation methods 安装方法

Standalone installer 独立安装程序

PyPI PyPI 的

Homebrew 自制

WinGet WinGet (温格)

Scoop 舀

Docker 码头工人

GitHub Releases GitHub 发布

Cargo 负责

Upgrading uv 升级紫外线

Shell autocompletion Shell 自动补全

Uninstallation 卸载

Next steps 后续步骤