

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

References 参考

Commands 命令

Guides 指南

Installing Python 安装 Python

If Python is already installed on your system, uv will **detect and use** it without configuration. However, uv can also install and manage Python versions. uv **automatically installs** missing Python versions as needed — you don't need to install Python to get started.

如果您的系统上已经安装了 Python，则 uv 将 无需配置即可**检测并使用**它。但是，uv 也可以安装和管理 Python 版本。uv 会根据需要**自动安装**缺少的 Python 版本——你无需安装 Python 即可开始使用。

Getting started 开始

To install the latest Python version:

要安装最新的 Python 版本：

```
$ uv python install
```

🔔 Note 注意

Python does not publish official distributable binaries. As such, uv uses distributions from the Astral [python-build-standalone](#) project. See the [Python distributions](#) documentation for more details.

Python 不发布官方的可分发二进制文件。因此，uv 使用来自 Astral [python-build-standalone](#) 项目的发行版。有关更多详细信息，请参阅 [Python 发行版文档](#)。

Once Python is installed, it will be used by `uv` commands automatically. uv also adds the installed version to your `PATH`:

安装 Python 后，它将自动被 `uv` 命令使用。uv 还会将已安装的版本添加到您的 `PATH` 中：

```
$ python3.13
```

uv only installs a **versioned** executable by default. To install `python` and `python3` executables, include the **experimental** `--default` option:

默认情况下，uv 仅安装版本化**可执行文件。要安装 `python` 和 `python3` 可执行文件，请包含**实验性** `--default` 选项：**

```
$ uv python install --default
```

💡 Tip 提示

See the documentation on [installing Python executables](#) for more details.
请参阅有关 [安装 Python 可执行文件](#) 的文档 了解更多详情。

Installing a specific version

安装特定版本

To install a specific Python version:

要安装特定的 Python 版本：

```
$ uv python install 3.12
```

To install multiple Python versions:

要安装多个 Python 版本：

```
$ uv python install 3.11 3.12
```

To install an alternative Python implementation, e.g., PyPy:

要安装替代的 Python 实现，例如 PyPy：

```
$ uv python install pypy3.10
```

See the [python `install`](#) documentation for more details.

有关更多详细信息，请参阅 [python 安装](#) 文档。

Reinstalling Python 重新安装 Python

To reinstall uv-managed Python versions, use `--reinstall`, e.g.:

要重新安装 uv 管理的 Python 版本，请使用 `--reinstall`，例如：

```
$ uv python install --reinstall
```

This will reinstall all previously installed Python versions. Improvements are constantly being added to the Python distributions, so reinstalling may resolve bugs even if the Python version does not change.

这将重新安装所有以前安装的 Python 版本。Python 发行版不断添加改进，因此即使 Python 版本没有更改，重新安装也可能会解决错误。

Viewing Python installations

查看 Python 安装

To view available and installed Python versions:

要查看可用和已安装的 Python 版本，请执行以下作：

```
$ uv python list
```

See the [python `list`](#) documentation for more details.

查看 [python 列表](#) 文档了解更多详情。

Automatic Python downloads

自动 Python 下载

Python does not need to be explicitly installed to use uv. By default, uv will automatically download Python versions when they are required. For example, the following would download Python 3.12 if it was not installed:

不需要显式安装 Python 即可使用 uv。默认情况下，uv 将在需要时自动下载 Python 版本。例如，如果未安装 Python 3.12，则以下内容将下载它：

```
$ uvx python@3.12 -c "print('hello world!')"
```

Even if a specific Python version is not requested, uv will download the latest version on demand. For example, if there are no Python versions on your system, the following will install Python before creating a new virtual environment:

即使没有请求特定的 Python 版本，uv 也会按需下载最新版本。例如，如果您的系统上没有 Python 版本，则在创建新的虚拟环境之前，将安装以下内容：

```
$ uv venv
```

💡 Tip 提示

Automatic Python downloads can be **easily disabled** if you want more control over when Python is downloaded.
如果您想更好地控制 Python 的下载时间，可以轻松**禁用**自动 Python 下载。

Using existing Python versions

使用现有的 Python 版本

uv will use existing Python installations if present on your system. There is no configuration necessary for this behavior: uv will use the system Python if it satisfies the requirements of the command invocation. See the [Python `discovery`](#) documentation for details.

uv 将使用现有的 Python 安装（如果存在于您的系统上）。没有配置 此行为所必需的：如果 uv 满足 命令调用。请参阅 有关详细信息，[请参阅 Python 发现文档](#)。

To force uv to use the system Python, provide the `--no-managed-python` flag. See the [Python `version preference`](#) documentation for more details.

要强制 uv 使用系统 Python，请提供 `--no-managed-python` 标志。请参阅 [Python 版本首选项](#) 文档了解更多详情。

Upgrading Python versions

升级 Python 版本

🔔 Important 重要

Support for upgrading Python patch versions is in *preview*. This means the behavior is experimental and subject to change.
对升级 Python 修补程序版本的支持处于**预览阶段**。这意味着该行为是实验性的，可能会发生变化。

To upgrade a Python version to the latest supported patch release:

要将 Python 版本升级到最新的受支持补丁版本，请执行以下作：

```
$ uv python upgrade 3.12
```

To upgrade all uv-managed Python versions:

要升级所有 uv 管理的 Python 版本：

```
$ uv python upgrade
```

See the [python `upgrade`](#) documentation for more details.

有关更多详细信息，请参阅 [python 升级](#) 文档。

Next steps 后续步骤

To learn more about `uv python`, see the [Python `version concept`](#) page and the [command reference](#).

要了解有关 `uv python` 的更多信息，请参阅 [Python 版本概念](#) 页面和[命令引用](#)。

Or, read on to learn how to [run scripts](#) and invoke Python with uv.

或者，继续阅读以了解如何[运行脚本](#)并使用 uv 调用 Python。

Getting started 开始

Installing a specific version
安装特定版本

Reinstalling Python
重新安装 Python

Viewing Python installations
查看 Python 安装

Automatic Python downloads
自动 Python 下载

Using existing Python versions
使用现有的 Python 版本

Upgrading Python versions
升级 Python 版本

Next steps 后续步骤