```
[![logo](../../assets/logo-letter.svg)](../.. "uv")
uν
Installation
Initializing search
[ uv ](https://github.com/astral-sh/uv "Go to repository")
[![logo](../../assets/logo-letter.svg)](../.. "uv") uv
[ uv ](https://github.com/astral-sh/uv "Go to repository")
 * [ Introduction ](../..)
 * [ Getting started ](../)
Getting started
  * Installation [ Installation ](./) Table of contents
    * Installation methods
     * Standalone installer
     * PyPI
     * Cargo
     * Homebrew
```

Skip to content

- \* WinGet
- \* Scoop \* Docker
- \* GitHub Releases
- \* Upgrading uv
- \* Shell autocompletion
- \* Uninstallation
- \* Next steps
- \* [ First steps ](../first-steps/)
- \* [ Features ](../features/)
- \* [ Getting help ](../help/)
- \* [ Guides ](../../guides/)

### Guides

- \* [ Installing Python ](../../guides/install-python/)
- \* [ Running scripts ](../../guides/scripts/)
- \* [ Using tools ](../../guides/tools/)
- \* [ Working on projects ](../../guides/projects/)
- \* [ Publishing packages ](../../guides/package/)
- \* [ Integrations ](../../guides/integration/)

## Integrations

- \* [ Docker ](../../guides/integration/docker/)
- \* [ Jupyter ](../../guides/integration/jupyter/)
- \* [ GitHub Actions ](../../guides/integration/github/)

```
* [ GitLab CI/CD ](../../guides/integration/gitlab/)
   * [ Pre-commit ](../../guides/integration/pre-commit/)
    * [ PyTorch ](../../guides/integration/pytorch/)
    * [ FastAPI ](../../guides/integration/fastapi/)
   * [ Alternative indexes ](../../guides/integration/alternative-indexes/)
    * [ Dependency bots ](../../guides/integration/dependency-bots/)
   * [ AWS Lambda ](../../guides/integration/aws-lambda/)
 * [ Concepts ](../../concepts/)
Concepts
  * [ Projects ](../../concepts/projects/)
Projects
   * [ Structure and files ](../../concepts/projects/layout/)
   * [ Creating projects ](../../concepts/projects/init/)
    * [ Managing dependencies ](../../concepts/projects/dependencies/)
   * [ Running commands ](../../concepts/projects/run/)
   * [ Locking and syncing ](../../concepts/projects/sync/)
   * [ Configuring projects ](../../concepts/projects/config/)
   * [ Building distributions ](../../concepts/projects/build/)
    * [ Using workspaces ](../../concepts/projects/workspaces/)
  * [ Tools ](../../concepts/tools/)
  * [ Python versions ](../../concepts/python-versions/)
  * [ Resolution ](../../concepts/resolution/)
```

\* [ Caching ](../../concepts/cache/)

```
* [ Configuration ](../../configuration/)
```

# Configuration

- \* [ Configuration files ](../../configuration/files/)
- \* [ Environment variables ](../../configuration/environment/)
- \* [ Authentication ](../../configuration/authentication/)
- \* [ Package indexes ](../../configuration/indexes/)
- \* [ Installer ](../../configuration/installer/)
- \* [ The pip interface ](../../pip/)

## The pip interface

- \* [ Using environments ](../../pip/environments/)
- \* [ Managing packages ](../../pip/packages/)
- \* [Inspecting packages ](../../pip/inspection/)
- \* [ Declaring dependencies ](../../pip/dependencies/)
- \* [ Locking environments ](../../pip/compile/)
- \* [ Compatibility with pip ](../../pip/compatibility/)
- \* [ Reference ](../../reference/)

#### Reference

- \* [ Commands ](../../reference/cli/)
- \* [ Settings ](../../reference/settings/)
- \* [ Troubleshooting ](../../reference/troubleshooting/)

## Troubleshooting

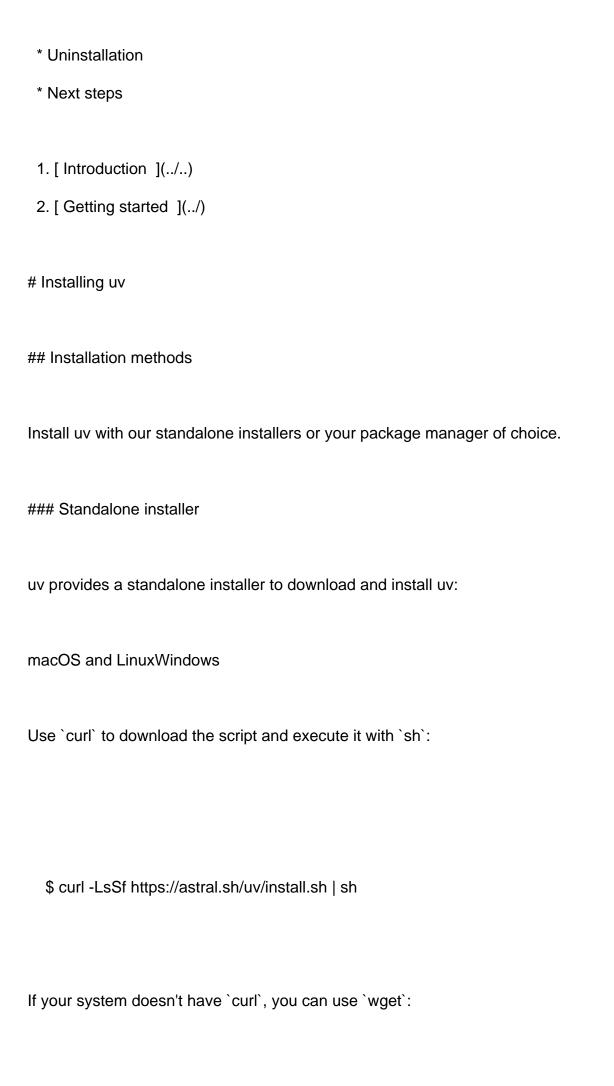
- \* [ Build failures ](../../reference/troubleshooting/build-failures/)
- \* [ Reproducible examples ](../../reference/troubleshooting/reproducible-examples/)
- \* [ Resolver ](../../reference/resolver-internals/)
- \* [ Benchmarks ](../../reference/benchmarks/)
- \* [ Policies ](../../reference/policies/)

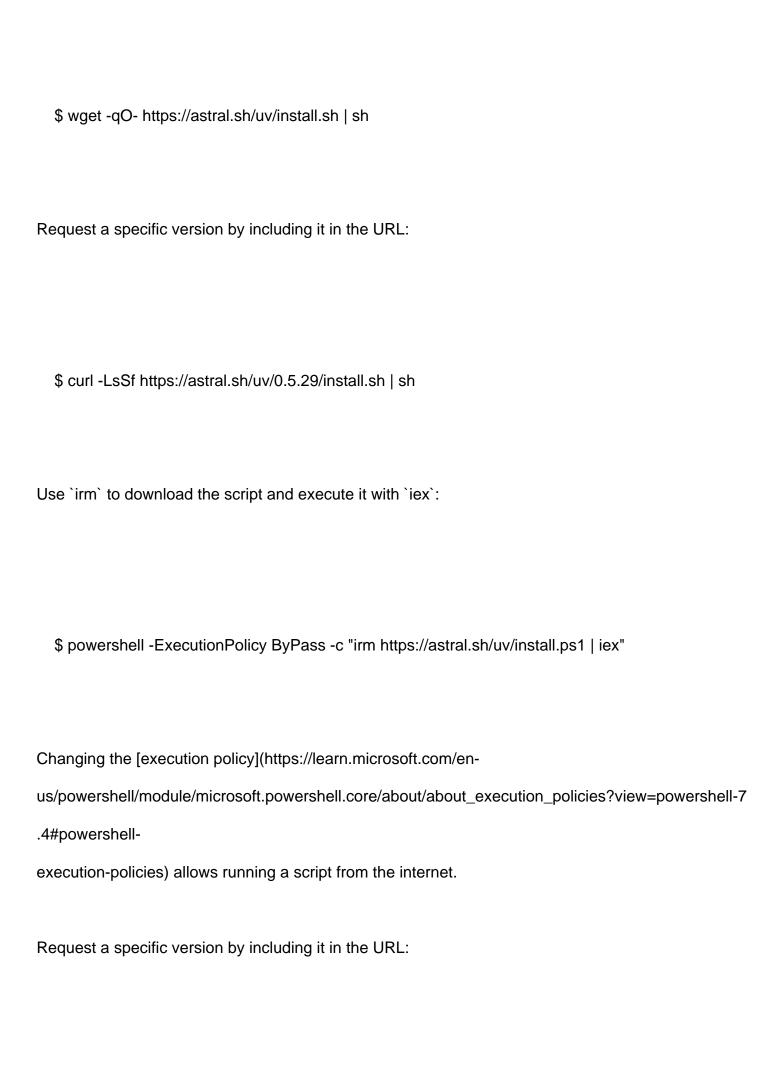
### **Policies**

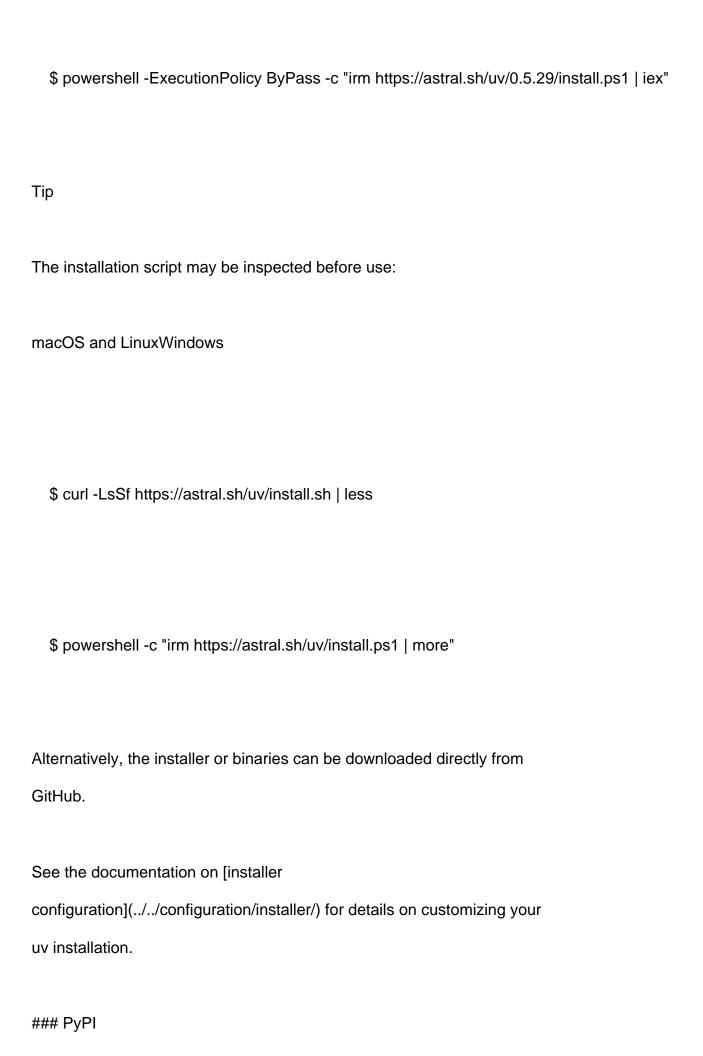
- \* [ Versioning ](../../reference/policies/versioning/)
- \* [ Platform support ](../../reference/policies/platforms/)
- \* [ License ](../../reference/policies/license/)

#### Table of contents

- \* Installation methods
  - \* Standalone installer
  - \* PyPI
- \* Cargo
- \* Homebrew
- \* WinGet
- \* Scoop
- \* Docker
- \* GitHub Releases
- \* Upgrading uv
- \* Shell autocompletion







For convenience, uv is published to [PyPI](https://pypi.org/project/uv/).

If installing from PyPI, we recommend installing uv into an isolated environment, e.g., with `pipx`:

However, 'pip' can also be used:

\$ pip install uv

Note

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](https://github.com/astral-sh/uv/blob/main/CONTRIBUTING.md#setup) for details on building uv from source.

### Cargo



uv is available via [Scoop](https://scoop.sh/#/apps?q=uv).

\$ scoop install main/uv

### Docker

uv provides a Docker image at [`ghcr.io/astral-sh/uv`](https://github.com/astral-sh/uv/pkgs/container/uv).

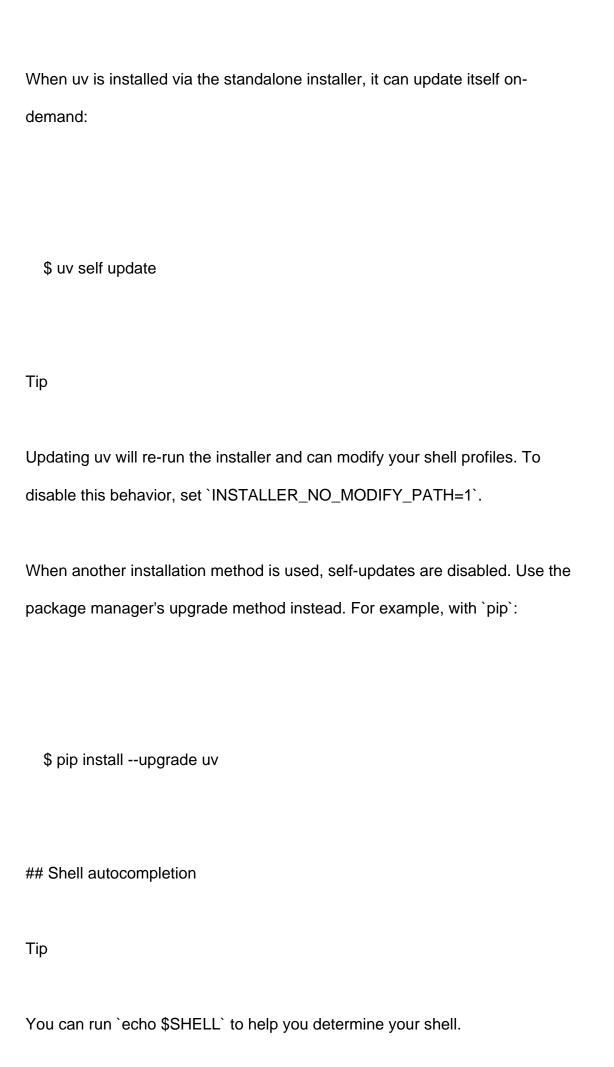
See our guide on [using uv in Docker](../../guides/integration/docker/) for more details.

### GitHub Releases

uv release artifacts can be downloaded directly from [GitHub Releases](https://github.com/astral-sh/uv/releases).

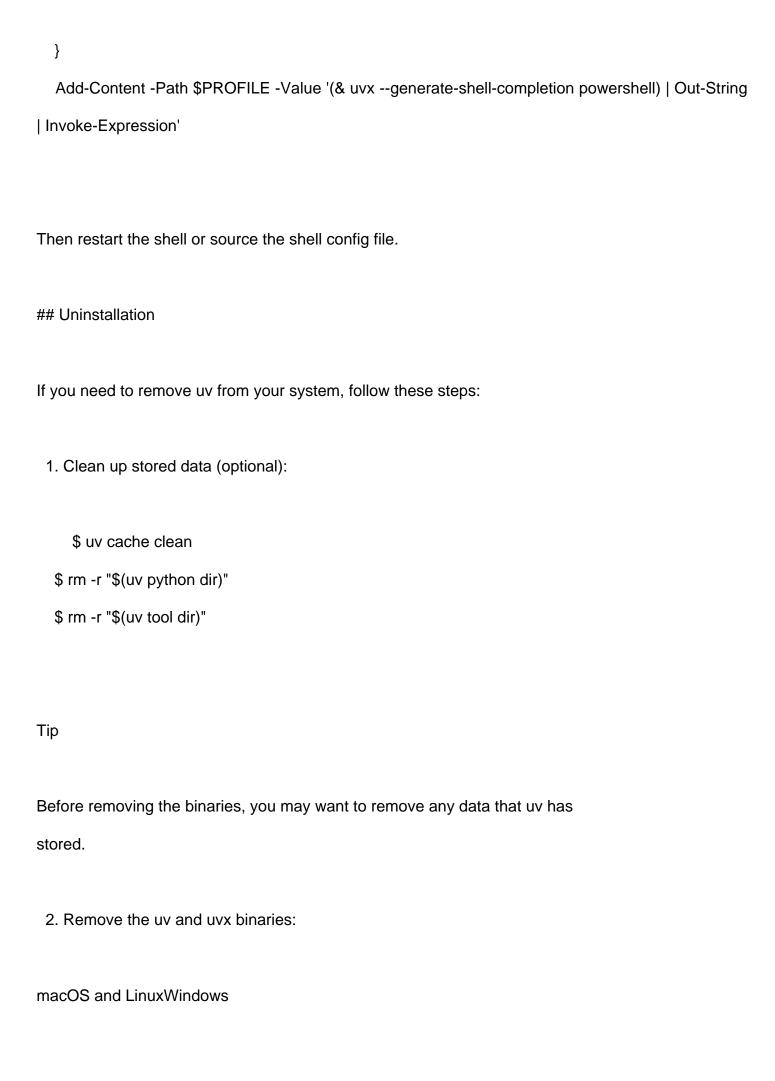
Each release page includes binaries for all supported platforms as well as instructions for using the standalone installer via `github.com` instead of `astral.sh`.

## Upgrading uv



To enable shell autocompletion for uv commands, run one of the following: BashZshfishElvishPowerShell / pwsh echo 'eval "\$(uv generate-shell-completion bash)" >> ~/.bashrc echo 'eval "\$(uv generate-shell-completion zsh)"' >> ~/.zshrc echo 'uv generate-shell-completion fish | source' >> ~/.config/fish/config.fish echo 'eval (uv generate-shell-completion elvish | slurp)' >> ~/.elvish/rc.elv if (!(Test-Path -Path \$PROFILE)) { New-Item -ItemType File -Path \$PROFILE -Force } Add-Content -Path \$PROFILE -Value '(& uv generate-shell-completion powershell) | Out-String |





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

\$ rm \$HOME\.local\bin\uv.exe

\$ rm \$HOME\.local\bin\uvx.exe

Note

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`.

## Next steps

See the [first steps](../first-steps/) or jump straight to the [guides](../../guides/) to start using uv.

February 6, 2025

Back to top [ Previous Index ](../) [ Next First steps ](../first-steps/)

Made with [ Material for MkDocs Insiders ](https://squidfunk.github.io/mkdocs-material/)

[](https://github.com/astral-sh/uv "github.com") [

```
](https://discord.com/invite/astral-sh "discord.com") [
](https://pypi.org/project/uv/ "pypi.org") [ ](https://x.com/astral_sh
"x.com")
```