

Installing `uv` on a Windows machine with administrative rights

The following procedure for installing `uv` is preferred, but only works if you have **administrative rights** on your machine.

Should you not have administrative rights, please refer to the instructions [here](#).

1. Open a PowerShell terminal

2. Install `uv` using the official installer script in PowerShell:

```
# run from an Administrator or normal PowerShell session
powershell -ExecutionPolicy Bypass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

3. Test if you can run `uv`:

```
uv --version
```

4. If this fails, try one of the following options:

4.1 Let `uv` update your shell automatically with:

```
uv tool update-shell
```

4.2 Alternatively: add it for the current session with:

```
$env:Path = "$env:USERPROFILE\.local\bin;$env:Path"
```

5. Try again to run `uv`

```
uv --version
```

6. Run the following to test everything

Note: If you are using conda, you should first deactivate any active conda environment (use `conda deactivate`).

The following series of terminal commands temporarily creates a project directory, creates a virtual environment, and installs a package, then removes everything again and goes back to the original directory you started in.

In the PowerShell, run:

```
# store the current working directory in a variable:
$env:OLDPWD = Get-Location
mkdir $HOME/test__uv -f
cd $HOME/test__uv
uv init --python=3.13
uv venv
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
.\.venv\Scripts\activate
uv add ipykernel
uv sync
cd $HOME
# delete the project recursively
Remove-Item -Recurse -Force test__uv
cd $env:OLDPWD
```