

Developer Life Hacks

Hidden Features, Pro Tips, and Time-Saving Tricks

Jonathan Agustin

December 7, 2024

GitHub Hacks

- Press `.` for instant web editor
- Use GitHub CLI `gh` for PRs/issues/merges in terminal
- Minimal context switching, improved flow

GitHub Demo

No video available

VS Code Essentials

- **Ctrl** + **X**: Delete line
- **Ctrl** + **/**: Toggle comments
- **Ctrl** + **Shift** + **P**: Command palette
- **Ctrl** + **P**: Quick file navigation
- **Alt** + **Click**: Multiple cursors

VS Code Demo

No video available

AI Assistance with Copilot

- Copilot: AI code suggestions
- `Ctrl` + `I`: Natural language refactoring
- Less grunt work, more architecture

Copilot Demo

No video available

Google Colab

- Pros:
 - No local setup required
 - Free GPU/TPU support
 - Easy to share and collaborate
- Cons / Tradeoffs:
 - Session timeouts require reruns
 - Limited persistence and customization
 - Potential slower file I/O for large datasets
- Ideal for rapid prototyping and teaching

GitHub Codespaces

- Pros:
 - Fully customizable dev environment in the cloud
 - Integrated with GitHub repos, PRs, issues
 - Consistent environment for team onboarding
- Cons / Tradeoffs:
 - Cost beyond free tier
 - Requires robust network connection
 - Not always as fast as local development for heavy tasks
- Great for standardized setups and scaling dev environments

uv - A Python Package Manager in Rust

- Pros:
 - Deterministic dependency resolution
 - Faster performance from Rust implementation
 - Lockfile ensures reproducible environments
- Cons / Tradeoffs:
 - Less widely adopted than pip/conda
 - Smaller community and fewer online resources
 - Might not support all Python ecosystem features yet
- Ideal if you value reproducibility and performance

Using uv

Installation: If available via a package manager (e.g., Homebrew on macOS):

Or download binaries from the [GitHub releases page](#).

Initialize a project:

This creates a `pyproject.toml` and a `uv.lock` file.

Adding dependencies:

Installing all dependencies:

`pipx` - Isolated Python CLI Tools

- Installs Python CLI tools in isolated venvs
- Avoids dependency/version conflicts with system Python
- Easy to upgrade or remove tools without breaking environments

Using pipx

Installation:

Install a CLI tool (e.g. Black):

Upgrade or uninstall:

Environment Management

- pyenv/conda/pipx: Manage Python versions/deps
- direnv: Auto-load environment variables per project
- Consistent, reliable setups

Environment/direnv Demo (30s)

No video available

Task & Taskfile

- Taskfile.yml: YAML-based tasks
- Dependencies for seamless workflows
- One-stop commands for build/test/deploy

Taskfile Example

```
n+ntversionp:+w l+s3
n+nttasksp:
+w  n+ntbuildp:
+w    n+ntdescp:+w l+lScalar+lScalarPlainBuild the project
+w    n+ntcmdsp:
+w      p+pIndicator+w l+lScalar+lScalarPlainpip install r requirements.tx
+w  n+nttestp:
+w    n+ntdescp:+w l+lScalar+lScalarPlainRun tests
+w    n+ntdepsp:+w p+pIndicator[n+nvbuildp+pIndicator]
+w    n+ntcmdsp:
+w      p+pIndicator+w l+lScalar+lScalarPlainpytest tests/
```

Task Demo

No video available

Code Quality Automation

- Black & Flake8: style & lint checks
- pre-commit hooks: checks at commit time
- High standards from the start

Code Quality Demo

No video available

New Slide Title

- New point 1
- New point 2
- New point 3

Your DX Superpowers

- GitHub & GH CLI
- VS Code & Copilot
- Google Colab & GitHub Codespaces
- pyenv/conda/pipx/uv & direnv
- Task & pre-commit

Resources & Next Steps

- [GitHub CLI Docs](#)
- [VS Code Documentation](#)
- [GitHub Copilot Info](#)
- [Google Colab Docs](#)
- [GitHub Codespaces](#)
- `pyenv/conda/pipx/uv/direnv/task/pre-commit docs`