GitHub Copilot CLI

Your terminal's new sidekick

Date: February 15, 2026

Public Preview

What is GitHub Copilot CLI?

- AI-powered coding assistant for your terminal
- Reads, writes, and runs code where you work
- Agent-powered with GitHub-native context
- Part of Copilot Pro, Pro+, Business, and Enterprise

Key Advantages

Less Setup

Included with your Copilot subscription—no extra costs or complex configuration

GitHub Native

Direct access to your repositories, issues, and pull requests from terminal

Full Control

Explicit approvals for every action—complete transparency and security

What Can Copilot CLI Do?

- **Explore projects** – Understand codebase structure instantly
- **Install dependencies** – Setup new projects in seconds
- **Edit files** – Make changes with conversational commands
- **Run commands** – Execute build, test, and deploy tasks
- **Debug issues** – Iterate fast without leaving terminal
- **Access GitHub context** – Link issues and PRs directly

Use Cases

Legacy Codebases

- Navigate unfamiliar code
- Understand dependencies
- Explain architecture

Cross-Platform Setup

- Configure dev environments
- Install toolchains
- Resolve compatibility

Multi-Step Tasks

- Refactor across files
- Build complex features
- Automate workflows

Rapid Prototyping

- Generate boilerplate
- Test ideas quickly
- Iterate locally

Extending with MCP Servers

MCP (Model Context Protocol) lets you extend Copilot CLI's capabilities:

- Connect custom tools and APIs
- Add domain-specific context
- Integrate internal systems
- Browse the GitHub MCP Registry

```
# Example: Search GitHub issues from terminal
gh copilot "Find beginner-friendly issues in cli/cli"
```

Frequently Asked Questions

Q: Who can access Copilot CLI?
A: Included with Copilot Pro, Pro+, Business, and Enterprise plans

Q: What operating systems are supported?
A: macOS, Linux, and Windows (via WSL)

Q: How much does it cost?
A: No additional cost—included in your Copilot subscription

Q: What models does CLI use?
A: Currently Claude Sonnet 4+ and GPT-5 (premium models only)

CLI vs VS Code Extension

Copilot CLI

- Terminal-focused workflow
- Agent-powered task execution
- Better for console operations
- Premium models only
- Explicit approval for actions

VS Code Extension

- Editor-integrated experience
- Inline code suggestions
- More mature feature set
- Free + premium models available
- Code checkpoints and richer UI

Community Feedback

Better Console Handling

"CLI is somewhat better at using console—extension tends to run tests in watch mode and timeout"

Complementary Tools

"Default to CLI if it solves your problems, use extension when it fails"

Different Approaches

"Different team, different system prompt leads to different characteristics"

Getting Started

Installation:

```
# Install GitHub CLI first (if not already installed)
brew install gh

# Install Copilot CLI extension
gh extension install github/gh-copilot

# Start using it
gh copilot
```

First Steps:

1. Run `gh copilot` to start interactive mode
2. Ask questions about your codebase
3. Request changes with natural language
4. Review and approve each action

Try GitHub Copilot CLI

Code faster, smarter, together.

Install Now · Documentation · Pricing

Questions?