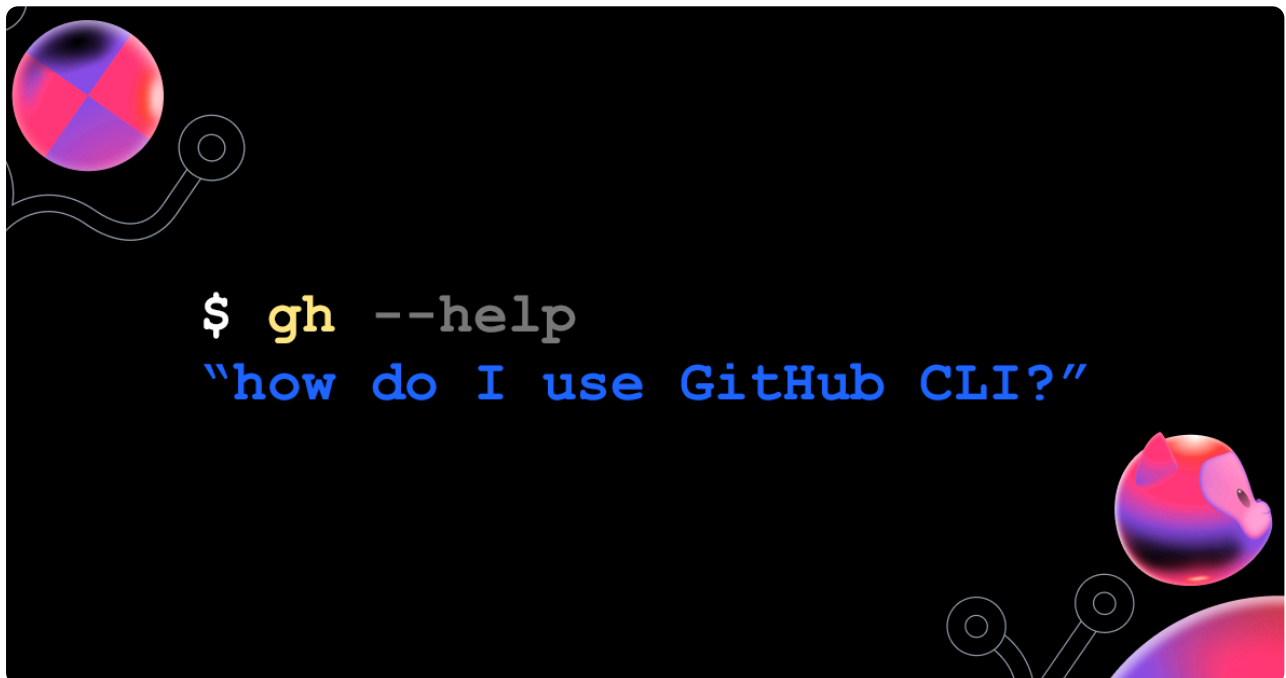

/ Blog

Home / Developer skills / GitHub

How to level up your Git game with GitHub CLI

Using Git in the CLI can improve your development speed and power. Here are our top eight commands for using GitHub via your command line.



Michelle Duke

Updated on August 17, 2024

🕒 14 minutes

Share:



Developers love working with the tools that make sense for them. At GitHub, we understand developers want to use GitHub in various ways. This includes using GitHub from the web, or even having a lightweight web editor with GitHub.dev. Developers can experience a full editing environment on the internet with Codespaces, use GitHub integrations with various editors, such as VS Code or JetBrains, and even use [GitHub](#)

/ Blog

GitHub and the terminal

We recently showed you many [Git commands you can use directly from your terminal](#). These commands are key, and every developer will use them when it comes to version control for your project. Not everyone, however, uses Git in the CLI. Doing so can improve your development speed and power. This is where you can become a power user of GitHub, using GitHub commands in the terminal alongside Git commands.

The [GitHub CLI](#) brings GitHub to your terminal. It allows you to have your entire GitHub workflow available from the command line. No more context switching, and more time staying in the flow: instead of working with Git on the command line and having to switch to a browser to perform GitHub related tasks, such as opening issues, reviewing pull requests, or editing repositories, you can do everything directly from your terminal. That means more time coding, and doing the thing you love ❤️.

If you're already familiar with GitHub, and you have either a Windows, Mac, or Linux operating system (OS), then this guide is for you. Here's how to get started with the GitHub CLI, and some of the top commands you can use for managing your projects in the way that suits you best.

How to get the GitHub CLI

To get started, download the GitHub CLI. Follow the instructions to download for [Windows](#), [Mac](#), and [Linux](#) on the [GitHub CLI](#) repository, or the open source [documentation](#). Once you've followed the instructions to install the GitHub CLI, open a terminal, and check to see if you have GitHub CLI by running `gh` to check. If you are on Windows, then you might receive something like this if you have the GitHub CLI:

```
PS C:\Users\miche> gh
Work seamlessly with GitHub from the command line.
```

Want to know if you have the latest version of GitHub CLI? You can type `gh --version` to see what version you have installed:

/ Blog

Once the CLI is installed, you'll need to authenticate with GitHub in order to continue. You can authenticate with GitHub by running the `gh auth login` command. You can follow the steps in [GitHub CLI Docs](#) to ensure you authenticate with the correct credentials.

Top 8 most useful GitHub CLI commands

These commands will be your bread and butter as a developer. You will use these almost every time you are working on a project.

Note that I'm working on a Windows machine, so the examples shown here will be from Windows PowerShell.

1. Create and clone repositories in your local environment

We use optional cookies to improve your experience on our websites and to display personalized advertising based on your online activity. If you reject optional cookies, only cookies necessary to provide you the services listed above will be used. You may change your selection on which cookies to accept by clicking "Manage Cookies" at the bottom of the page to change your selection. This selection is maintained

Accept

Reject

Manage cookies

for 180 days.
Please review
your
selections
regularly.

[How to
manage
cookie
preferences](#) |
[Privacy
Statement](#) |
[Third-Party
Cookies](#).

authentication and Git configuration, making your workflow faster.

Other useful commands for working with repositories include:

- `gh repo list` which will include a [list of all repositories](#) that are owned by that username.
- `gh repo edit <OWNER/REPO>` to [edit the settings of a particular repository](#).
- `gh repo delete <OWNER/REPO>` to [delete a particular repository](#).

There are lots of other useful commands for working with repositories, and you can [view the full list](#), including relevant flags, in GitHub CLI Docs.

2. Quickly view, create, and edit issues

Now that you have a new repository in your development environment, you can use the terminal to work on the project. Let's start with issues. [Issues](#) are like tracking tasks for your project. They might be for creating a new feature, or fixing a bug in the code.

By looking at the list of issues you have, you can pick one to work on, or see an overview view of the current work assigned to the project. Use the `gh issue list` command to see a [list of your currently open issues](#):

```
PS C:\Users\miche\OneDrive\Documents\git-repos\cli>
```

To see this list, you'll need to make sure your terminal is pointed towards the repository you want to work in. For example `.\cli` in the above demonstration or `.\my-awesome-project` for the new project repository we created in the first demo.

/ Blog

```
PS C:\Users\mimanner>
```

You can then [edit issues](#) further by using `gh issue edit` or even try launching the notepad whilst you are creating the issue to add the body text for that issue.


Another useful command is `gh issue status` which shows the [current status of all the open issues in a repository](#), and how they relate to you:

```
PS C:\Users\mimanner> |
```

There are lots more useful commands for working with issues and you can find the full list in [GitHub CLI Docs](#).

/ Blog

this development work from the terminal. There are several useful commands when it comes to working with pull requests. Once you've navigated to your repository, you can [view a list of open pull requests](#) by typing `gh pr list` into your terminal. You'll receive a list of all the open pull requests in your repository. This will also include any draft pull requests:



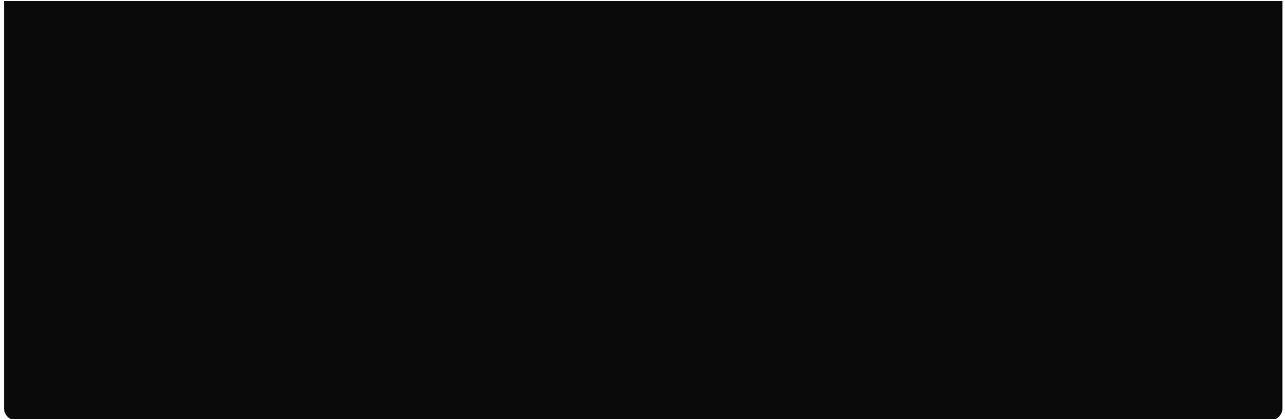
```
PS C:\Users\miche\OneDrive\Documents\git-repos\cli>
```

Note that the pull requests shown in gray are draft pull requests. Draft pull requests are pull requests that we can create when we're actively working on changes, but aren't quite ready to merge those changes. Draft pull requests mean we can start collaborating without our code being completely polished. This is a new feature and you can read more about draft pull requests in our blog post.

You might also like to create a pull request directly from the terminal. We can use a combination of Git and GitHub commands to achieve this. For example, `git checkout` will switch you to the branch of your repository. Then, you can use `gh pr create` to [create a pull request](#). You can then follow the prompts on the terminal to open the pull request.

Here's an example of checking the list of pull requests. Use `git checkout` to ensure we're on the correct branch that has the changes to merge. I can then use `gh pr create` to create the pull request. By following the prompts, I'm able to open a notepad to add a body for the pull request. Since I haven't tested these changes, I want to create this particular pull request as a draft pull request. Finally, I can use `gh pr list` to see a list of the repository's pull requests with the new draft pull request I just created:

/ Blog



Here are some other useful commands to use in relation to pull requests:

- `gh pr status` to see the [status of PRs](#) and how they relate to you.
- `gh pr checkout` to [switch to a particular branch](#) and see information for that branch. This is super powerful. As a reviewer, you can now quickly pull down the branch and test the changes locally before approving or merging a pull request.
- `gh pr merge` to [merge the pull request](#).

[View the full list of pull request related issues](#), including relevant flags, in GitHub CLI Docs.

4. Create and view releases for your project

When working on a project, you might like to work with the releases for that project. A release is a software package and includes release notes and binary files for other people to use. You can create a new release by using `gh release create` and follow the prompts to create a release for your project:

/ Blog



Use `gh release list` to [view releases for a particular project](#), and use `gh release view` to view a project's particular release.

5. Monitor your GitHub Actions workflows at a glance

No project is complete without [GitHub Actions](#), and now you can manage and run your GitHub Actions from the command line. GitHub Actions is our automation platform, allowing you to automate your project from idea to production. You can build, test, and deploy your project directly from GitHub using GitHub Actions. If you get confused around what is an action vs. what is a workflow, [check out our guide](#) on the difference between a GitHub Actions, actions, and workflows.

Use `gh workflow list` to view a [list of all your available workflows](#).



This list shows the workflows that the repository owner has set up. Each project will have its own type and number of workflows. Once you can see the full list of workflows available within a particular project's repository, use `gh workflow run` to [select and run a particular workflow](#).

For example, if you wanted to run the “code scanning” action above, you can use `gh workflow run 1208059` to run that workflow. You'll need to have admin rights to a

/ Blog

workflow, and see the runs of that workflow:

```
PS C:\Users\miche\OneDrive\Documents\git-repos\mish-friendly-food>
```

6. Write and edit labels

Labels are an important part of effectively managing projects. They help color-code and group particular issues so we can triage and manage them correctly. You can see a list of all the labels for your repository by typing `gh label list`. Then use `gh label create` to [add a new label](#), or `gh label edit` to [edit a particular label](#):

```
PS C:\Users\mimanner> cd "C:\Users\mimanner\OneDrive - Microsoft\Documents\git-repos\my-awesome-project"
PS C:\Users\mimanner\OneDrive - Microsoft\Documents\git-repos\my-awesome-project> |
```


7. See your GitHub status

One of the most useful commands is understanding your [GitHub status](#). Use `gh status` and you'll be provided with a list of GitHub status as it pertains to you. This includes your assigned issues and pull requests, any open pull requests that require your review, and any other repository activity:

/ Blog

8. Add more to your GitHub CLI with extensions

We've shown you lots of useful commands for working with GitHub. But what if there's something you want to do that isn't mentioned here? Since the GitHub CLI is open source, many developers—both at GitHub and in the community—[create extensions for the GitHub CLI](#). These extensions add additional features to your command line. You can browse all the [available GitHub CLI extensions](#) via the terminal using `gh ext browse` and read a brief overview of each extension does:



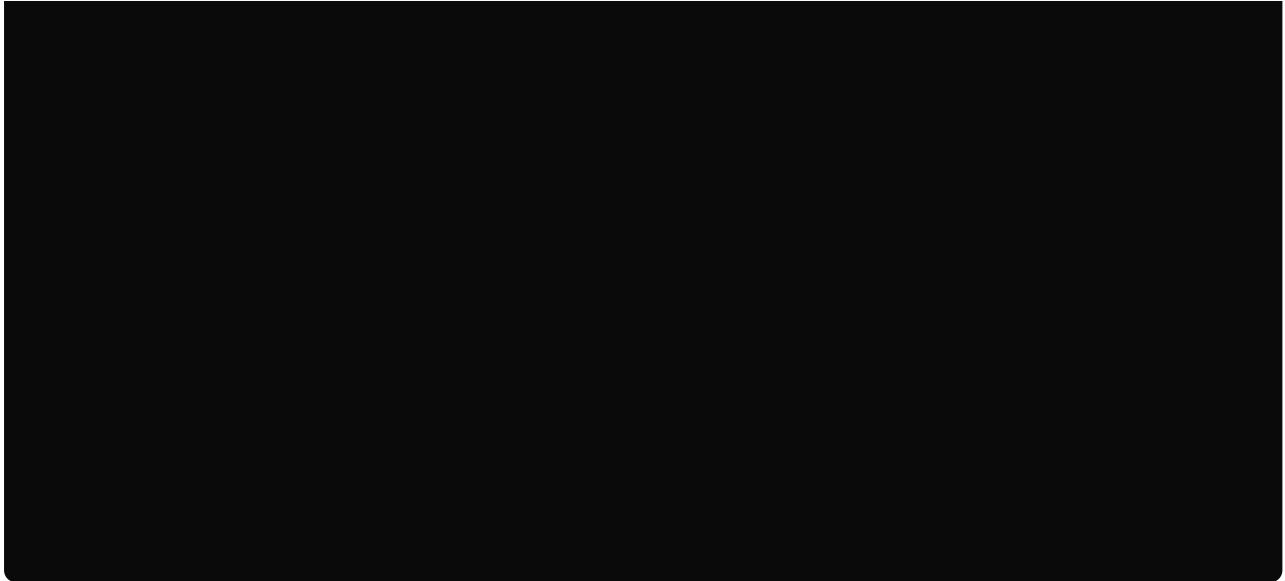
```
PS C:\Users\miche> gh ext browse
```

If there's an extension that you'd like to add that functionality for your use, you can install it by using `gh ext install <OWNER/REPO> .`

Bonus. Help?!

What happens if you can't remember a particular command? Use `gh help` to see a list of helpful commands:

/ Blog



What's next?

Now we've walked through how to use GitHub CLI from your terminal, start practicing some of these commands. If you love working in the terminal, then this will help reduce context switching and you can spend more time in the place you love—the command line.

So, what happens when you get stuck in the terminal? Yes, you can use `gh help` to get some useful suggestions on where to go next, but it's often not enough, or not exactly what we're looking for. If this is you, then GitHub is here to help. In our next blog post, we'll teach you how to use [GitHub Copilot in the CLI](#) to get unstuck when you run into errors or can't remember a particular command. Stay tuned.

Tags :

GitHub CLI

Written by



Michelle Duke

[@MishManners](#)

I'm a Content Producer working in tech & innovation. Known as the "Hackathon Queen" 🏆 I'm on the GitHub DevRel team and love sharing

/ Blog

More on GitHub CLI

Where does your software (really) come from?

GitHub is working with the OSS community to bring new supply chain security capabilities to the platform.

Trevor Rosen

Calling all teachers! Learn how to build new commands on the GitHub Classroom CLI

In this step-by-step tutorial, we'll dive into how you can become the next open source contributor to the GitHub Classroom CLI, building commands that you can use to improve your workflow as an educator!

Zenara Daley

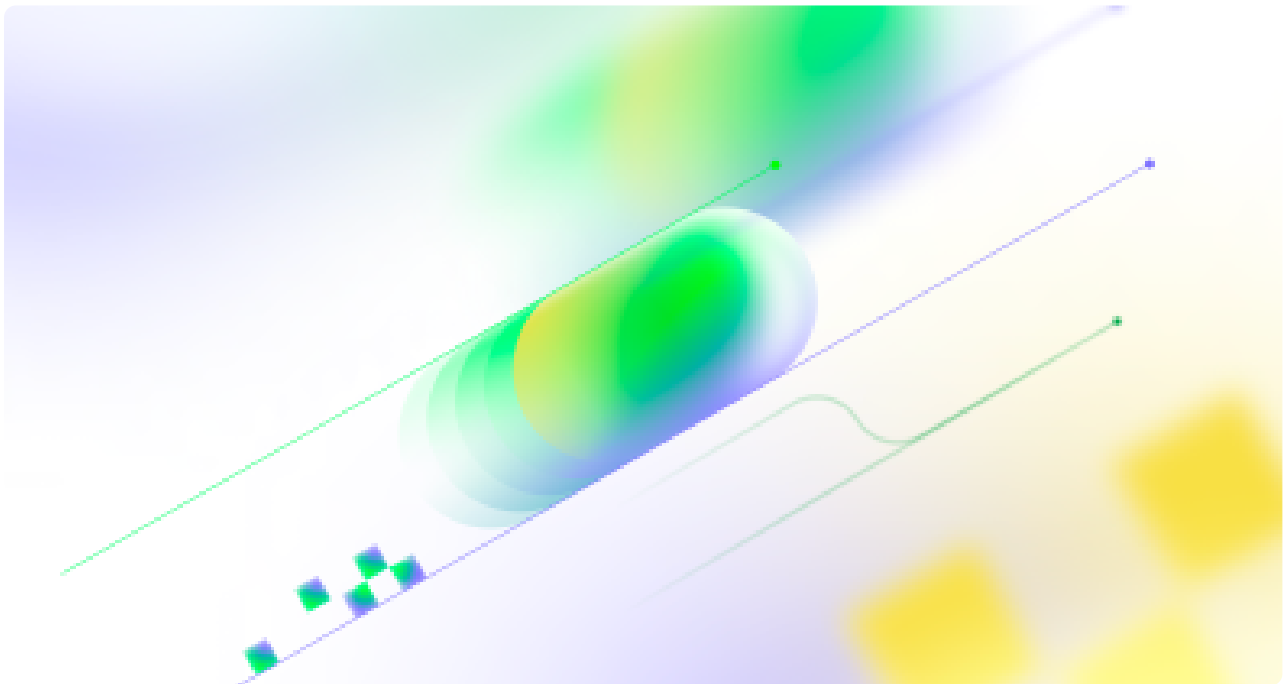
Related posts

[/ Blog](#)

Beginner's guide to GitHub: Creating a pull request

As part of the GitHub for Beginners guide, learn how to create pull requests. This will enable you to suggest changes to existing repositories.

Kedasha Kerr



How to use AI coding tools to learn a new programming language

Explore how AI coding tools like GitHub Copilot can accelerate your journey to learn new programming languages.

[/ Blog](#)

Beginner's guide to GitHub: Adding code to your repository

Take the next step in our GitHub for Beginners series and add code to your repository. Learn how to create branches and upload changes into a pull request.

Kedasha Kerr

Explore more from GitHub



Docs

Everything you need to master GitHub, all in one place.

[Go to Docs](#) 

/ Blog

Build what's next on GitHub, the place for anyone from anywhere to build anything.

Start building >



Customer stories

Meet the companies and engineering teams that build with GitHub.

Learn more [↗](#)



GitHub Universe 2024

Get tickets to the 10th anniversary of our global developer event on AI, DevEx, and security.

Get tickets [↗](#)

Subscribe to our newsletter

Code with confidence. Discover tips, technical guides, and best practices in our biweekly newsletter just for devs.

Your email address

☐ Yes please, I'd like GitHub and affiliates to use my information for personalized communications, targeted advertising and campaign effectiveness. See the [GitHub Privacy Statement](#) for more details.



Product	Platform	Support	Company
Features	Developer API	Docs	About
Security	Partners	Community Forum	Blog
Enterprise	Atom	Training	Careers
Customer Stories	Electron	Status	Press
Pricing	GitHub Desktop	Contact	Shop
Resources			

/ Blog
