

Connecting to GitHub

In this article

Introduction

Getting started

Making more complex changes in the browser

Working from the desktop

Comparison of tools for connecting to GitHub

Further reading

Learn about the tools you can use to work with repositories hosted on GitHub.

Introduction

GitHub is a web-based app that lets you host files in repositories, collaborate on work, and track changes to files over time. Version tracking on GitHub is powered by the open source software Git. Whenever you update a repository on GitHub, Git tracks the changes you make.

There are many ways to work with GitHub, and you can choose a method that suits your level of experience, personal preferences, and the repositories you work with. For example, you can choose whether you want to work in the browser or from your desktop, how you want to use Git, and what capabilities you need from your editor and other software. You may choose to work with different repositories in different ways.

If you're new to GitHub, a good way to start contributing is to make changes in the browser on GitHub.com. As you become more familiar with GitHub and start contributing larger changes, you may want to start working with other tools. This article explains how to progress through these stages and helps you choose the best tool for your requirements at each stage. To quickly compare all the tools available for working with GitHub, see "[Comparison of tools for connecting to GitHub](#)."

Getting started

In the user interface on GitHub.com, you can perform the whole "GitHub flow" for contributing to a repository, including creating a branch or fork, editing and previewing files, committing your changes, and creating a pull request. You can also upload files from your computer or download them from the repository. For more information, see "[GitHub flow](#)."

Working directly on GitHub.com is often the quickest way to contribute to a repository, for the following reasons.


- You're working directly with the repository hosted on GitHub, so you don't have to download a copy of the repository to your computer and keep this copy in sync.
- If you're already signed in to GitHub, you have access to any repository where you have the necessary permissions, so you don't need to set up any additional authentication on your computer.

- You can commit changes in the user interface, so you don't need to use the command line or memorize any Git commands.

For a tutorial to help you get started with making changes in the browser, see "[Hello World](#)."

Making more complex changes in the browser

Working directly on GitHub.com is best for small, simple changes, often targeting a single file in a repository. If you want to work in the browser but need to make more complex changes, such as moving content between files, you can choose from the following tools to open a repository in a dedicated editor.

- If you want an editor where you can quickly open or create files, you can press the  key in any repository to open the github.dev editor. This is a lightweight web-based editor that includes many of the features of Visual Studio Code, such as a search bar and buttons for Git commands. For more information, see "[The github.dev web-based editor](#)."
- If you want to stay in the browser but need to do things like run commands, create a test build of your project, or install dependencies, you can open a repository in a codespace. A codespace is a remote development environment with storage and compute power. It includes an editor and integrated terminal, and comes preinstalled with common tools you may need to work with a project, including Git. For more information, see "[GitHub Codespaces overview](#)."

Alternatively, you can connect to GitHub from your desktop, and work with a local copy of the repository.

Working from the desktop

To work with a repository from your desktop, you'll need to download (or "clone") a copy of the repository to your computer, then push any changes you make to GitHub. Working from your desktop can have several advantages over working in the browser.

- You can work with all your local files and tools.
- You have access to compute power. For example, you might need to run a script to create a local preview of a site, so you can test the changes you're making.
- You don't need an Internet connection to work on a project.

If you haven't worked with a GitHub repository from your desktop before, you'll need to authenticate to GitHub from your computer, so you can access the repositories you need. You may also need to set up your working environment with the tools you need to contribute, such as Git, an editor, and dependencies for a project. For these reasons, it can take longer to get started if you want to work from your desktop, compared to working in the browser.

There are several tools you can use to connect to GitHub from your desktop. These tools allow you to authenticate to GitHub, clone a repository, track your changes, and push the changes to GitHub.

- If you want a lot of control and flexibility, you can use the command line. You'll need to install Git and be familiar with some basic Git commands. You can also install GitHub CLI, a command-line interface that lets you perform many actions on GitHub, such as creating a pull request or forking a repository. For more information, see "[Set up Git](#)" and "[About GitHub CLI](#)."
- If you'd prefer to use a visual interface, you can use a visual Git client such as GitHub Desktop. With GitHub Desktop, you can visualize the changes you're making and access most Git commands through a visual interface, so you don't need to memorize any commands. For more information, see "[About GitHub Desktop](#)."

- If you want to work in one place, you can often do most things from your editor. An editor such as VS Code includes an integrated terminal and buttons for common Git commands, so you can edit files and push your changes to GitHub all from one place. You can also install an extension to work directly with pull requests and issues on GitHub. To get started, see [Download VS Code](#).

Comparison of tools for connecting to GitHub

The following table provides a comparison between the tools you can use to work with repositories on GitHub, both in your browser and from your desktop.

You can perform the whole GitHub flow with any of the tools described here: each tool includes access not only to Git commands for tracking the changes you've made, but also to GitHub-specific features, so you can create a pull request with your proposed changes from wherever you're working.

For more information about how to authenticate to GitHub Enterprise Cloud with each of these tools, see "[About authentication to GitHub](#)."

Tool	Use case	Browser or desktop
On GitHub.com	You want a visual interface and need to make quick, simple changes, typically involving a single commit. For an introduction, see " Hello World ."	Browser
github.dev	You want to make more complex changes to a repository than is possible on GitHub.com, but don't need to work with a terminal or tools you have installed on your computer. For more information, see " The github.dev web-based editor ."	Browser
GitHub Codespaces	You need the resources of a computer to do things like run scripts, create a test build of your project, or install dependencies, and you want to get started quickly by working in a cloud-based environment. For more information, see " GitHub Codespaces overview ."	Browser or desktop
GitHub Desktop	You want to work with files locally, and would prefer a visual interface to use Git, visualize changes, and interact with GitHub Enterprise Cloud. For more information, see " About GitHub Desktop ."	Desktop
IDE or text editor	You're working with more complex files and projects and want everything in one place.	Desktop
Command-line Git and GitHub CLI	You're used to working from the command line and want to avoid switching context, or you	Desktop

need to access a complex Git command that isn't integrated into visual interfaces. For more information, see "[Set up Git](#)" and "[About GitHub CLI](#)."

GitHub API

You want to automate common tasks such as backing up your data, or create integrations that extend GitHub. For more information, see "[About GitHub's APIs](#)."

Browser or desktop

Further reading

- "[About Git](#)"

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)