GitHub Introduction

Contents

[Creating a GitHub Account 5](#_Toc164293476)

[Getting started with your GitHub account 6](#_Toc164293477)

[In this article 6](#_Toc164293478)

[Part 1: Configuring your GitHub account 6](#_Toc164293479)

[1. Creating an account 6](#_Toc164293480)

[2. Choosing your GitHub product 6](#_Toc164293481)

[3. Verifying your email address 6](#_Toc164293482)

[4. Configuring two-factor authentication 7](#_Toc164293483)

[5. Viewing your GitHub profile and contribution graph 7](#_Toc164293484)

[Part 2: Using GitHub's tools and processes 7](#_Toc164293485)

[1. Learning Git 7](#_Toc164293486)

[2. Setting up Git 7](#_Toc164293487)

[3. Choosing how to interact with GitHub 8](#_Toc164293488)

[4. Writing on GitHub 8](#_Toc164293489)

[5. Searching on GitHub 8](#_Toc164293490)

[6. Managing files on GitHub 8](#_Toc164293491)

[Part 3: Collaborating on GitHub 9](#_Toc164293492)

[1. Working with repositories 9](#_Toc164293493)

[2. Importing your projects 9](#_Toc164293494)

[3. Managing collaborators and permissions 10](#_Toc164293495)

[4. Managing repository settings 10](#_Toc164293496)

[5. Setting up your project for healthy contributions 10](#_Toc164293497)

[6. Using GitHub Issues and Projects 10](#_Toc164293498)

[7. Managing notifications 10](#_Toc164293499)

[8. Working with GitHub Pages 11](#_Toc164293500)

[9. Using GitHub Discussions 11](#_Toc164293501)

[Part 4: Customizing and automating your work on GitHub 11](#_Toc164293502)

[1. Using GitHub Marketplace 11](#_Toc164293503)

[2. Using the GitHub API 11](#_Toc164293504)

[3. Building GitHub Actions 11](#_Toc164293505)

[4. Publishing and managing GitHub Packages 12](#_Toc164293506)

[Part 5: Building securely on GitHub 12](#_Toc164293507)

[1. Securing your repository 12](#_Toc164293508)

[2. Managing your dependencies 12](#_Toc164293509)

[Part 6: Participating in GitHub's community 12](#_Toc164293510)

[1. Contributing to open source projects 13](#_Toc164293511)

[2. Interacting with GitHub Community Support 13](#_Toc164293512)

[3. Reading about GitHub on GitHub Docs 13](#_Toc164293513)

[4. Learning with GitHub Skills 13](#_Toc164293514)

[5. Supporting the open source community 13](#_Toc164293515)

[6. Contacting GitHub Support 13](#_Toc164293516)

[Creating a GitHub Pages site 15](#_Toc164293517)

[Who can use this feature? 16](#_Toc164293518)

[In this article 16](#_Toc164293519)

[Creating a repository for your site 16](#_Toc164293520)

[Creating your site 18](#_Toc164293521)

[Next steps 19](#_Toc164293522)

[Git Bash 21](#_Toc164293523)

[Git Bash | What is it & How to Use it 22](#_Toc164293524)

[Git Bash Topics Covered 22](#_Toc164293525)

[Introduction to Git Bash 22](#_Toc164293526)

[Git Bash Download 23](#_Toc164293527)

[Using Git Bash with GitHub 23](#_Toc164293528)

[Git Bash Commands 23](#_Toc164293529)

[Using GitHub with GitKraken Client 25](#_Toc164293530)

[Set Git Bash as Your Default Terminal in GitKraken Client 26](#_Toc164293531)

[Git Bash FAQ 26](#_Toc164293532)

[Q: How to Paste into Git Bash? 26](#_Toc164293533)

[Q: How to Update Git Bash? 26](#_Toc164293534)

[GitHub Cheat Sheet 27](#_Toc164293535)

[SETUP 28](#_Toc164293536)

[SETUP & INIT 28](#_Toc164293537)

[STAGE & SNAPSHOT 28](#_Toc164293538)

[BRANCH & MERGE 28](#_Toc164293539)

[INSPECT & COMPARE 28](#_Toc164293540)

[TRACKING PATH CHANGES 29](#_Toc164293541)

[IGNORING PATTERNS 29](#_Toc164293542)

[SHARE & UPDATE 29](#_Toc164293543)

[REWRITE HISTORY 29](#_Toc164293544)

[TEMPORARY COMMITS 29](#_Toc164293545)

[GitHub Learning Lab 30](#_Toc164293546)

[GitHub App 31](#_Toc164293547)

[Getting started with GitHub Learning Lab 31](#_Toc164293548)

[Recommended approach: Install GitHub Learning Lab on all repositories 31](#_Toc164293549)

[Alternative approach: Install GitHub Learning Lab on a single repository 31](#_Toc164293550)

# Creating a GitHub Account

Documentation Source: [GitHub Getting Started](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account)

## Getting started with your GitHub account

With a personal account on GitHub, you can import or create repositories, collaborate with others, and connect with the GitHub community.

## In this article

This guide will walk you through setting up your GitHub account and getting started with GitHub's features for collaboration and community.

## [Part 1: Configuring your GitHub account](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#part-1-configuring-your-github-account)

The first steps in starting with GitHub are to create an account, choose a product that fits your needs best, verify your email, set up two-factor authentication, and view your profile.

There are several types of accounts on GitHub. Every person who uses GitHub has their own personal account, which can be part of multiple organizations and teams. Your personal account is your identity on GitHub.com and represents you as an individual.

### [1. Creating an account](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#1-creating-an-account)

To sign up for an account on GitHub.com, navigate to [**https://github.com/**](https://github.com/) and follow the prompts.

To keep your GitHub account secure you should use a strong and unique password. For more information, see "[**Creating a strong password**](https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-strong-password)."

### [2. Choosing your GitHub product](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#2-choosing-your-github-product)

You can choose GitHub Free or GitHub Pro to get access to different features for your personal account. You can upgrade at any time if you are unsure at first which product you want.

For more information on all of GitHub's plans, see "[**GitHub’s plans**](https://docs.github.com/en/get-started/learning-about-github/githubs-plans)."

### [3. Verifying your email address](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#3-verifying-your-email-address)

To ensure you can use all the features in your GitHub plan, verify your email address after signing up for a new account. For more information, see "[**Verifying your email address**](https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-personal-account-on-github/managing-email-preferences/verifying-your-email-address)."

### [4. Configuring two-factor authentication](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#4-configuring-two-factor-authentication)

Two-factor authentication, or 2FA, is an extra layer of security used when logging into websites or apps. We strongly urge you to configure 2FA for the safety of your account. For more information, see "[**About two-factor authentication**](https://docs.github.com/en/authentication/securing-your-account-with-two-factor-authentication-2fa/about-two-factor-authentication)."

Optionally, after you have configured 2FA, add a passkey to your account to enable a secure, passwordless login. For more information, see "[**About passkeys**](https://docs.github.com/en/authentication/authenticating-with-a-passkey/about-passkeys)" and "[**Managing your passkeys**](https://docs.github.com/en/authentication/authenticating-with-a-passkey/managing-your-passkeys)."

### [5. Viewing your GitHub profile and contribution graph](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#5-viewing-your-github-profile-and-contribution-graph)

Your GitHub profile tells people the story of your work through the repositories and gists you've pinned, the organization memberships you've chosen to publicize, the contributions you've made, and the projects you've created. For more information, see "[**About your profile**](https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-profile/customizing-your-profile/about-your-profile)" and "[**Viewing contributions on your profile**](https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-profile/managing-contribution-settings-on-your-profile/viewing-contributions-on-your-profile)."

## [Part 2: Using GitHub's tools and processes](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#part-2-using-githubs-tools-and-processes)

To best use GitHub, you'll need to set up Git. Git is responsible for everything GitHub-related that happens locally on your computer. To effectively collaborate on GitHub, you'll write in issues and pull requests using GitHub Flavored Markdown.

### [1. Learning Git](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#1-learning-git)

GitHub's collaborative approach to development depends on publishing commits from your local repository to GitHub for other people to view, fetch, and update using Git. For more information about Git, see the "[**Git Handbook**](https://guides.github.com/introduction/git-handbook/)" guide. For more information about how Git is used on GitHub, see "[**GitHub flow**](https://docs.github.com/en/get-started/using-github/github-flow)."

### [2. Setting up Git](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#2-setting-up-git)

If you plan to use Git locally on your computer, whether through the command line, an IDE or text editor, you will need to install and set up Git. For more information, see "[**Set up Git**](https://docs.github.com/en/get-started/getting-started-with-git/set-up-git)."

If you prefer to use a visual interface, you can download and use GitHub Desktop. GitHub Desktop comes packaged with Git, so there is no need to install Git separately. For more information, see "[**Getting started with GitHub Desktop**](https://docs.github.com/en/desktop/overview/getting-started-with-github-desktop)."

Once you install Git, you can connect to GitHub repositories from your local computer, whether your own repository or another user's fork. When you connect to a repository on GitHub.com from Git, you'll need to authenticate with GitHub using either HTTPS or SSH. For more information, see "[**About remote repositories**](https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories)."

### [3. Choosing how to interact with GitHub](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#3-choosing-how-to-interact-with-github)

Everyone has their own unique workflow for interacting with GitHub; the interfaces and methods you use depend on your preference and what works best for your needs.

For more information about the different approaches for interacting with GitHub, and a comparison of the tools you can use, see "[**Connecting to GitHub**](https://docs.github.com/en/get-started/using-github/connecting-to-github)."

### [4. Writing on GitHub](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#4-writing-on-github)

To make your communication clear and organized in issues and pull requests, you can use GitHub Flavored Markdown for formatting, which combines an easy-to-read, easy-to-write syntax with some custom functionality. For more information, see "[**About writing and formatting on GitHub**](https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/about-writing-and-formatting-on-github)."

You can learn GitHub Flavored Markdown with the "[**Communicate using Markdown**](https://github.com/skills/communicate-using-markdown)" course on GitHub Skills.

### [5. Searching on GitHub](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#5-searching-on-github)

Our integrated search allows you to find what you are looking for among the many repositories, users and lines of code on GitHub. You can search globally across all of GitHub or limit your search to a particular repository or organization. For more information about the types of searches you can do on GitHub, see "[**About searching on GitHub**](https://docs.github.com/en/search-github/getting-started-with-searching-on-github/about-searching-on-github)."

Our search syntax allows you to construct queries using qualifiers to specify what you want to search for. For more information on the search syntax to use in search, see "[**Searching on GitHub**](https://docs.github.com/en/search-github/searching-on-github)."

### [6. Managing files on GitHub](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#6-managing-files-on-github)

With GitHub, you can create, edit, move and delete files in your repository or any repository you have write access to. You can also track the history of changes in a file line by line. For more information, see "[**Managing files**](https://docs.github.com/en/repositories/working-with-files/managing-files)."

## [Part 3: Collaborating on GitHub](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#part-3-collaborating-on-github)

Any number of people can work together in repositories across GitHub. You can configure settings, create projects, and manage your notifications to encourage effective collaboration.

### [1. Working with repositories](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#1-working-with-repositories)

#### [Creating a repository](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#creating-a-repository)

A repository is like a folder for your project. You can have any number of public and private repositories in your personal account. Repositories can contain folders and files, images, videos, spreadsheets, and data sets, as well as the revision history for all files in the repository. For more information, see "[**About repositories**](https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories)."

When you create a new repository, you should initialize the repository with a README file to let people know about your project. For more information, see "[**Creating a new repository**](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-new-repository)."

#### [Cloning a repository](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#cloning-a-repository)

You can clone an existing repository from GitHub to your local computer, making it easier to add or remove files, fix merge conflicts, or make complex commits. Cloning a repository pulls down a full copy of all the repository data that GitHub has at that point in time, including all versions of every file and folder for the project. For more information, see "[**Cloning a repository**](https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository)."

#### [Forking a repository](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#forking-a-repository)

A fork is a copy of a repository that you manage, where any changes you make will not affect the original repository unless you submit a pull request to the project owner. Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea. For more information, see "[**Working with forks**](https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks)."

### [2. Importing your projects](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#2-importing-your-projects)

If you have existing projects you'd like to move over to GitHub you can import projects using the GitHub Importer, the command line, or external migration tools. For more information, see "[**Importing source code**](https://docs.github.com/en/migrations/importing-source-code)."

### [3. Managing collaborators and permissions](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#3-managing-collaborators-and-permissions)

You can collaborate on your project with others using your repository's issues, pull requests, and projects (classic). You can invite other people to your repository as collaborators from the **Collaborators** tab in the repository settings. For more information, see "[**Inviting collaborators to a personal repository**](https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-personal-account-on-github/managing-access-to-your-personal-repositories/inviting-collaborators-to-a-personal-repository)."

You are the owner of any repository you create in your personal account and have full control of the repository. Collaborators have write access to your repository, limiting what they have permission to do. For more information, see "[**Permission levels for a personal account repository**](https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-personal-account-on-github/managing-user-account-settings/permission-levels-for-a-personal-account-repository)."

### [4. Managing repository settings](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#4-managing-repository-settings)

As the owner of a repository you can configure several settings, including the repository's visibility, topics, and social media preview. For more information, see "[**Managing your repository’s settings and features**](https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features)."

### [5. Setting up your project for healthy contributions](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#5-setting-up-your-project-for-healthy-contributions)

To encourage collaborators in your repository, you need a community that encourages people to use, contribute to, and evangelize your project. For more information, see "[**Building Welcoming Communities**](https://opensource.guide/building-community/)" in the Open Source Guides.

By adding files like contributing guidelines, a code of conduct, and a license to your repository you can create an environment where it's easier for collaborators to make meaningful, useful contributions. For more information, see "[**Setting up your project for healthy contributions**](https://docs.github.com/en/communities/setting-up-your-project-for-healthy-contributions)."

### [6. Using GitHub Issues and Projects](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#6-using-github-issues-and-projects)

You can use GitHub Issues to organize your work with issues and pull requests and manage your workflow with Projects. For more information, see "[**About issues**](https://docs.github.com/en/issues/tracking-your-work-with-issues/about-issues)" and "[**About Projects**](https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects)."

### [7. Managing notifications](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#7-managing-notifications)

Notifications provide updates about the activity on GitHub you've subscribed to or participated in. If you're no longer interested in a conversation, you can unsubscribe, unwatch, or customize the types of notifications you'll receive in the future. For more information, see "[**About notifications**](https://docs.github.com/en/account-and-profile/managing-subscriptions-and-notifications-on-github/setting-up-notifications/about-notifications)."

### [8. Working with GitHub Pages](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#8-working-with-github-pages)

You can use GitHub Pages to create and host a website directly from a repository on GitHub.com. For more information, see "[**About GitHub Pages**](https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages)."

### [9. Using GitHub Discussions](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#9-using-github-discussions)

You can enable GitHub Discussions for your repository to help build a community around your project. Maintainers, contributors and visitors can use discussions to share announcements, ask and answer questions, and participate in conversations around goals. For more information, see "[**About discussions**](https://docs.github.com/en/discussions/collaborating-with-your-community-using-discussions/about-discussions)."

## [Part 4: Customizing and automating your work on GitHub](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#part-4-customizing-and-automating-your-work-on-github)

You can use tools from the GitHub Marketplace, the GitHub API, and existing GitHub features to customize and automate your work.

### [1. Using GitHub Marketplace](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#1-using-github-marketplace)

GitHub Marketplace contains integrations that add functionality and improve your workflow. You can discover, browse, and install free and paid tools, including GitHub Apps, OAuth apps, and GitHub Actions, in [**GitHub Marketplace**](https://github.com/marketplace).

### [2. Using the GitHub API](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#2-using-the-github-api)

There are two versions of the GitHub API: the REST API and the GraphQL API. You can use the GitHub APIs to automate common tasks, [**back up your data**](https://docs.github.com/en/repositories/archiving-a-github-repository/backing-up-a-repository), or [**create integrations**](https://docs.github.com/en/get-started/exploring-integrations/about-integrations) that extend GitHub. For more information, see "[**Comparing GitHub's REST API and GraphQL API**](https://docs.github.com/en/rest/overview/about-githubs-apis)."

### [3. Building GitHub Actions](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#3-building-github-actions)

With GitHub Actions, you can automate and customize GitHub.com's development workflow on GitHub. You can create your own actions, and use and customize actions shared by the GitHub community. For more information, see "[**Learn GitHub Actions**](https://docs.github.com/en/actions/learn-github-actions)."

### [4. Publishing and managing GitHub Packages](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#4-publishing-and-managing-github-packages)

GitHub Packages is a software package hosting service that allows you to host your software packages privately or publicly and use packages as dependencies in your projects. For more information, see "[**Introduction to GitHub Packages**](https://docs.github.com/en/packages/learn-github-packages/introduction-to-github-packages)."

## [Part 5: Building securely on GitHub](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#part-5-building-securely-on-github)

GitHub has a variety of security features that help keep code and secrets secure in repositories. Some features are available for all repositories, while others are only available for public repositories and repositories with a GitHub Advanced Security license. For an overview of GitHub security features, see "[**GitHub security features**](https://docs.github.com/en/code-security/getting-started/github-security-features)."

### [1. Securing your repository](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#1-securing-your-repository)

As a repository administrator, you can secure your repositories by configuring repository security settings. These include managing access to your repository, setting a security policy, and managing dependencies. For public repositories, and for private repositories owned by organizations where GitHub Advanced Security is enabled, you can also configure code and secret scanning to automatically identify vulnerabilities and ensure tokens and keys are not exposed.

For more information on steps you can take to secure your repositories, see "**[Quickstart for securing your repository](https://docs.github.com/en/code-security/getting-started/securing-your-repository)**."

### [2. Managing your dependencies](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#2-managing-your-dependencies)

A large part of building securely is maintaining your project's dependencies to ensure that all packages and applications you depend on are updated and secure. You can manage your repository's dependencies on GitHub by exploring the dependency graph for your repository, using Dependabot to automatically raise pull requests to keep your dependencies up-to-date, and receiving Dependabot alerts and security updates for vulnerable dependencies.

For more information, see "[**Securing your software supply chain**](https://docs.github.com/en/code-security/supply-chain-security)."

## [Part 6: Participating in GitHub's community](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#part-6-participating-in-githubs-community)

There are many ways to participate in the GitHub community. You can contribute to open source projects, interact with people in the GitHub Community Support, or learn with GitHub Skills.

### [1. Contributing to open source projects](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#1-contributing-to-open-source-projects)

Contributing to open source projects on GitHub can be a rewarding way to learn, teach, and build experience in just about any skill you can imagine. For more information, see "[**How to Contribute to Open Source**](https://opensource.guide/how-to-contribute/)" in the Open Source Guides.

You can find personalized recommendations for projects and good first issues based on your past contributions, stars, and other activities in [**Explore GitHub**](https://github.com/explore). For more information, see "[**Finding ways to contribute to open source on GitHub**](https://docs.github.com/en/get-started/exploring-projects-on-github/finding-ways-to-contribute-to-open-source-on-github)."

### [2. Interacting with GitHub Community Support](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#2-interacting-with-github-community-support)

You can connect with developers around the world to ask and answer questions, learn, and interact directly with GitHub staff. To get the conversation started, see "[**GitHub Community Support**](https://github.com/orgs/community/discussions/)."

### [3. Reading about GitHub on GitHub Docs](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#3-reading-about-github-on-github-docs)

You can read documentation that reflects the features available to you on GitHub. For more information, see "[**About versions of GitHub Docs**](https://docs.github.com/en/get-started/learning-about-github/about-versions-of-github-docs)."

### [4. Learning with GitHub Skills](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#4-learning-with-github-skills)

You can learn new skills by completing fun, realistic projects in your very own GitHub repository with [**GitHub Skills**](https://skills.github.com/). Each course is a hands-on lesson created by the GitHub community and taught by a friendly bot.

For more information, see "[**Git and GitHub learning resources**](https://docs.github.com/en/get-started/start-your-journey/git-and-github-learning-resources)."

### [5. Supporting the open source community](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#5-supporting-the-open-source-community)

GitHub Sponsors allows you to make a monthly recurring payment to a developer or organization who designs, creates, or maintains open source projects you depend on. For more information, see "[**About GitHub Sponsors**](https://docs.github.com/en/sponsors/getting-started-with-github-sponsors/about-github-sponsors)."

### [6. Contacting GitHub Support](https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#6-contacting-github-support)

GitHub Support can help you troubleshoot issues you run into while using GitHub. For more information, see "[**About GitHub Support**](https://docs.github.com/en/support/learning-about-github-support/about-github-support)."

# Creating a GitHub Pages site

Document source: [GitHub Pages](https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site)

You can create a GitHub Pages site in a new or existing repository.

## Who can use this feature?

GitHub Pages is available in public repositories with GitHub Free and GitHub Free for organizations, and in public and private repositories with GitHub Pro, GitHub Team, GitHub Enterprise Cloud, and GitHub Enterprise Server. For more information, see "[GitHub’s plans](https://docs.github.com/en/get-started/learning-about-github/githubs-plans)."

All GitHub Pages builds will use GitHub Actions from June 30, 2024. No other changes are required but GitHub Actions must be enabled in your repository for builds to continue. For more information on enabling GitHub Actions, see "[Managing GitHub Actions settings for a repository](https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/enabling-features-for-your-repository/managing-github-actions-settings-for-a-repository)."

## In this article

**Note:** Organization owners can restrict the publication of GitHub Pages sites from repositories owned by the organization. For more information, see "[**Managing the publication of GitHub Pages sites for your organization**](https://docs.github.com/en/organizations/managing-organization-settings/managing-the-publication-of-github-pages-sites-for-your-organization)."

## [Creating a repository for your site](https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site#creating-a-repository-for-your-site)

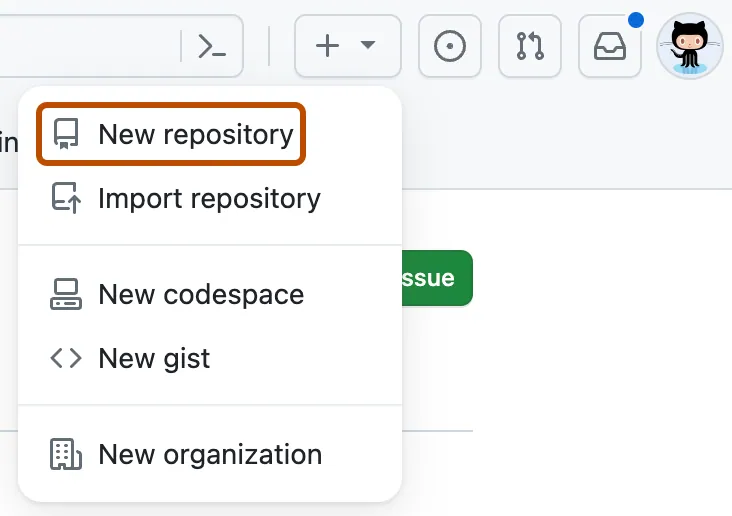
You can either create a repository or choose an existing repository for your site.

If you want to create a GitHub Pages site for a repository where not all of the files in the repository are related to the site, you will be able to configure a publishing source for your site. For example, you can have a dedicated branch and folder to hold your site source files, or you can use a custom GitHub Actions workflow to build and deploy your site source files.

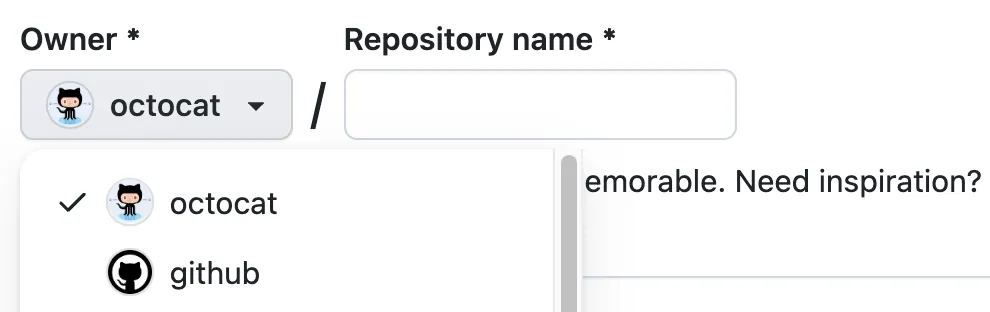
If the account that owns the repository uses GitHub Free or GitHub Free for organizations, the repository must be public.

If you want to create a site in an existing repository, skip to the "[**Creating your site**](https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site#creating-your-site)" section.

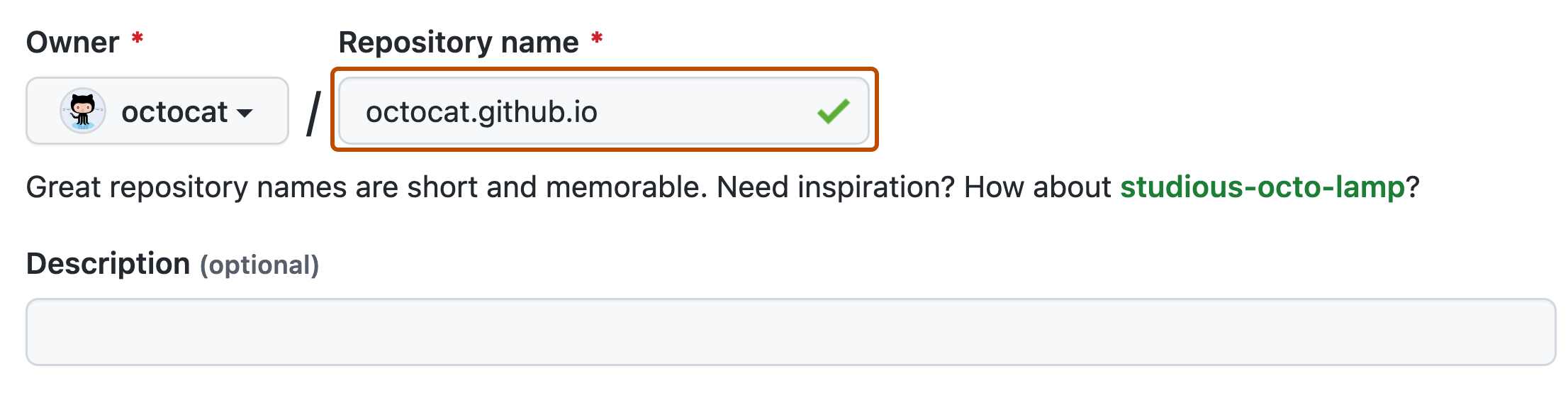
1. In the upper-right corner of any page, select , then click **New repository**.



1. Use the **Owner** dropdown menu to select the account you want to own the repository.



1. Type a name for your repository and an optional description. If you're creating a user or organization site, your repository must be named <user>.github.io or <organization>.github.io. If your user or organization name contains uppercase letters, you must lowercase the letters. For more information, see "[**About GitHub Pages**](https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages#types-of-github-pages-sites)."



1. Choose a repository visibility. For more information, see "[**About repositories**](https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories#about-repository-visibility)."
2. Select **Initialize this repository with a README**.
3. Click **Create repository**.

## [Creating your site](https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site#creating-your-site)

Before you can create your site, you must have a repository for your site on GitHub. If you're not creating your site in an existing repository, see "[**Creating a repository for your site**](https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site#creating-a-repository-for-your-site)."

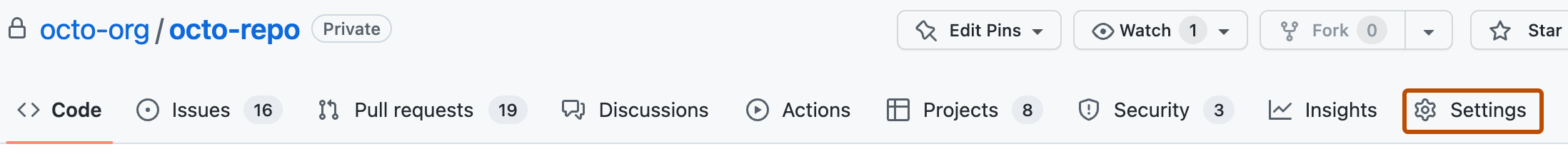
**Warning**: GitHub Pages sites are publicly available on the internet, even if the repository for the site is private. If you have sensitive data in your site's repository, you may want to remove the data before publishing. For more information, see "[**About repositories**](https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories#about-repository-visibility)."

1. On GitHub, navigate to your site's repository.
2. Decide which publishing source you want to use. For more information, see "[**Configuring a publishing source for your GitHub Pages site**](https://docs.github.com/en/pages/getting-started-with-github-pages/configuring-a-publishing-source-for-your-github-pages-site)."
3. Create the entry file for your site. GitHub Pages will look for an index.html, index.md, or README.md file as the entry file for your site.

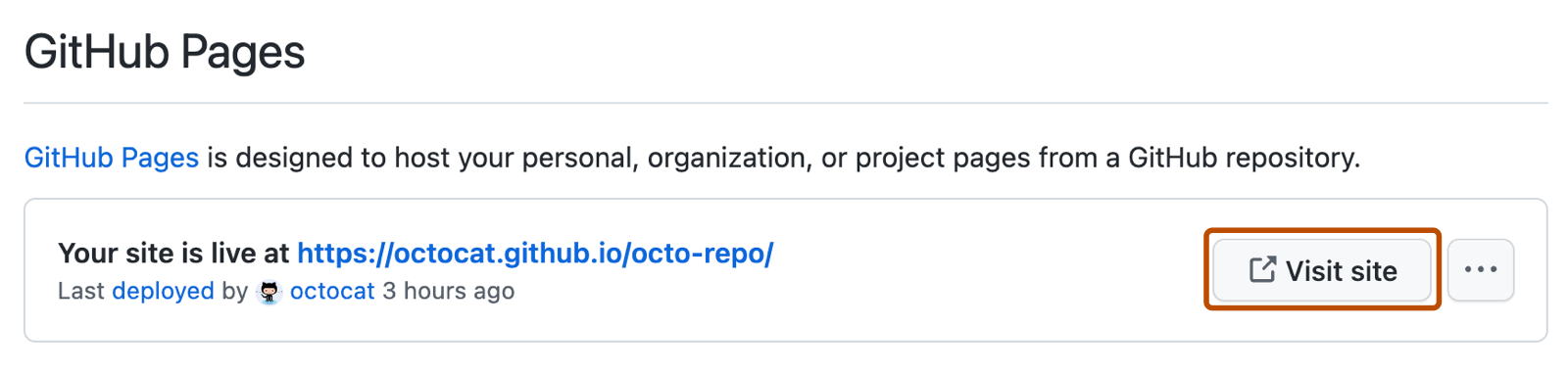
If your publishing source is a branch and folder, the entry file must be at the top level of the source folder on the source branch. For example, if your publishing source is the /docs folder on the main branch, your entry file must be located in the /docs folder on a branch called main.

If your publishing source is a GitHub Actions workflow, the artifact that you deploy must include the entry file at the top level of the artifact. Instead of adding the entry file to your repository, you may choose to have your GitHub Actions workflow generate your entry file when the workflow runs.

1. Configure your publishing source. For more information, see "[**Configuring a publishing source for your GitHub Pages site**](https://docs.github.com/en/pages/getting-started-with-github-pages/configuring-a-publishing-source-for-your-github-pages-site)."
2. Under your repository name, click **Settings**. If you cannot see the "Settings" tab, select the dropdown menu, then click **Settings**.



1. In the "Code and automation" section of the sidebar, click **Pages**.
2. To see your published site, under "GitHub Pages", click **Visit site**



**Note:** It can take up to 10 minutes for changes to your site to publish after you push the changes to GitHub. If you don't see your GitHub Pages site changes reflected in your browser after an hour, see "[**About Jekyll build errors for GitHub Pages sites**](https://docs.github.com/en/pages/setting-up-a-github-pages-site-with-jekyll/about-jekyll-build-errors-for-github-pages-sites)."

1. Your GitHub Pages site is built and deployed with a GitHub Actions workflow. For more information, see "[**Viewing workflow run history**](https://docs.github.com/en/actions/monitoring-and-troubleshooting-workflows/viewing-workflow-run-history)".

**Note:** GitHub Actions is free for public repositories. Usage charges apply for private and internal repositories that go beyond the monthly allotment of free minutes. For more information, see "[**Usage limits, billing, and administration**](https://docs.github.com/en/actions/learn-github-actions/usage-limits-billing-and-administration)".

**Notes**:

* If you are publishing from a branch and your site has not published automatically, make sure someone with admin permissions and a verified email address has pushed to the publishing source.
* Commits pushed by a GitHub Actions workflow that uses the GITHUB\_TOKEN do not trigger a GitHub Pages build.

## [Next steps](https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site#next-steps)

You can add more pages to your site by creating more new files. Each file will be available on your site in the same directory structure as your publishing source. For example, if the publishing source for your project site is the gh-pages branch, and you create a new file called /about/contact-us.md on the gh-pages branch, the file will be available at https://<user>.github.io/<repository>/about/contact-us.html.

You can also add a theme to customize your site’s look and feel. For more information, see "[**Adding a theme to your GitHub Pages site using Jekyll**](https://docs.github.com/en/pages/setting-up-a-github-pages-site-with-jekyll/adding-a-theme-to-your-github-pages-site-using-jekyll)".

To customize your site even more, you can use Jekyll, a static site generator with built-in support for GitHub Pages. For more information, see "[**About GitHub Pages and Jekyll**](https://docs.github.com/en/pages/setting-up-a-github-pages-site-with-jekyll/about-github-pages-and-jekyll)".

# Git Bash

Documentation Source: [gitkraken](https://www.gitkraken.com/blog/what-is-git-bash#introduction-to-git-bash)

## Git Bash | What is it & How to Use it

Put simply, [Git Bash](https://www.gnu.org/software/bash/) is an application for Microsoft Windows OS environments that provides Unix based shell utilities and experience for Git command line commands. Git Bash emulates the Git command line experience that Unix environments have, for Windows users. Most Windows users download Git Bash when they first [install Git for Windows](https://www.gitkraken.com/learn/git/git-download#download-git-for-windows).

As a version control system, [Git](https://git-scm.com/) was originally delivered within Unix style command line methods. MacOS & Linux Operating Systems have a built-in terminal shell that supports Unix-based command line features whereas Microsoft Windows’ Operating System command line prompt is not a Unix-based terminal.

Because the Windows command line does not support Unix-based commands, Git CLI features are mostly delivered with user-friendly GUI applications in the Windows Operating System.

These applications provide visual functionalities to the end-user which makes using Git easier. Even some of them, like [GitKraken Client](https://www.gitkraken.com/git-client), provide drag-and-drop functionalities for common [Git commands](https://www.gitkraken.com/learn/git/commands), so end-users don’t need to know every single command for managing their code base. This makes life easier for beginners at the very first stages of Git usage.

As Git experience increases, end-users can prefer using Git command line features for specific commands. Git Bash is one tool that provides command line features in the Windows Operating System to end-users. Another tool that developers can use to interact with the command line on Windows is [GitKraken Client’s CLI](https://www.gitkraken.com/cli).

### Git Bash Topics Covered

[Download GitKraken Client Free](https://www.gitkraken.com/download/windows64)

[Other Platforms](https://www.gitkraken.com/download)

## Introduction to Git Bash

A shell is a computer application that integrates with the operating system and exposes its services to an end-user or other applications. Bash is an acronym for Bourne Again Shell, which is the GNU Project’s shell.

Git Bash is not just a bash package for Microsoft Windows OS. It includes bash utilities, Unix collections like Secure Shell Protocol ([SSH](https://www.gitkraken.com/learn/git/tutorials/how-git-ssh-works)), Secure Copy Protocol ([SCP](https://en.wikipedia.org/wiki/Secure_copy_protocol)), [CAT](https://en.wikipedia.org/wiki/Cat_(Unix)) (Unix utility that reads files sequentially, writing them to standard output), and other Unix-based collections compiled for Windows and Git features.

Git Bash provides a package for Git usage from the command line for Windows users, but [GitKraken Client](https://www.gitkraken.com/git-client) will make those same actions faster and more intuitive.

Some things absent from Git Bash that users can enjoy in GitKraken Client are auto-suggest and auto-complete for Git commands. Just start typing a command in the terminal tab and you will see relevant command suggestions with descriptions. Simply select the command you’re looking for and hit enter to automagically complete the Git action!

## Git Bash Download

Git Bash can be installed as other Windows applications, but you need to first download the executable file from the Git Bash download page and then follow the installation steps.

1. Open your favorite browser and go to the [Git Bash Download](https://git-scm.com/downloads) page on Git-scm.

2. After successfully downloading the Git Bash executable file, follow the installation steps described in the below embedded video. Some hints are also listed in the video for the related installation step.

3. Voilà! Git Bash is ready to be used on your local Microsoft Windows platform.

\*If you used the above process to download Git Bash and Git for Windows for the first time, you may find instructions for [configuring Git after your Git download](https://www.gitkraken.com/learn/git/git-download#configure-git) useful. You’ll need to configure Git before you can access all of Git’s capabilities and features.

## Using Git Bash with GitHub

In this section, we will look at how to run Git Bash and see some basic Git Bash commands required for Git integration. As Git Bash is a command line utility for Git on the Microsoft Windows platform, a basic Microsoft Windows command prompt (CMD) knowledge will be useful before getting started with Git Bash as they are very similar. If you do not know how to use CMD features, you can just take a look at the [Windows Commands Reference](https://www.microsoft.com/en-us/download/details.aspx?id=56846) from the Microsoft related web site.

Now, let’s look at an example of using Git Bash with GitHub. You will start by linking your GitHub account with Git Bash to start configuring your GitHub repositories. If you do not have a GitHub account, you can create one directly from the [GitHub](https://github.com/) home page.

You can also refer to [GitHub Docs](https://docs.github.com/en) for a very large and illustrative documentation about GitHub usage.

## Git Bash Commands

Before going into steps on how to configure Git Bash and how to use it, you need to have a repository on GitHub. If this is your first time creating a repository on GitHub, checkout the related [GitHub documentation](https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/creating-a-repository-on-github) for instructions on how to create a repository on GitHub.

Now, let’s start configuring Git Bash with your GitHub account from scratch (examples in this article will be given from [my personal GitHub account](https://github.com/evrentan)).

1. First step is to run Git Bash. Double click the Git Bash icon on your Windows desktop to open your Git Bash interface.

2. Use the cd Git bash command to change your active directory with your local repository workspace. cd and chdir in Windows CMD are aliases for setting the active directory.

Now your active directory is your local repository. You can validate your active directory with the pwd command.

3. This step is related to configuring your GitHub email and GitHub username. Type the below commands to link your GitHub email & GitHub username.

git config --global user.name "%yourGitHubUserName%"

git config --global user.email "%yourGitHubUserEmail%

a. Now, you can clone the “git-bash-intro” repository to your local workspace. First, get the clone link from your GitHub repository as shown below:

b. Then, type the below command to clone your repository. You will use the clone link that you just copied from your GitHub repository.

There are two important tips while [cloning the Git repository](https://www.gitkraken.com/learn/git/git-clone). First: if you create a private repository, you also need to have related GitHub permissions. In this Git Bash example, the git-bash-intro repository was initialized as public. You can check the [repository visibility section](https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/creating-a-repository-on-github/about-repository-visibility) from GitHub Docs for more information about GitHub repository permissions.

Another helpful tip is that you may face the below error while cloning a repository:

fatal: could not create work tree dir 'git-bash-intro': permission denied

This error states that you do not start Git Bash with the required permission to execute changes on your local Microsoft Windows platform. You can run Git Bash as administrator to resolve.

4. Next, you will clone a repository to your local workspace to create a folder with the same name as your GitHub repository.

5. Add a new text file called “firstCommit.txt” in your git-bash-intro local repository.

6. It is time to reflect your local changes to your remote GitHub repository. Here, you will need to run a set of commands to push changes to your remote git-bash-intro repository.

git add .

git commit -m "first commit"

git push origin master

Now, let’s check what these Git bash commands mean:

* git add: This command adds content to the git index and updates the index with the local working tree. You can use “.” to add all working tree contents, or you can explicitly mention which content you want to add.For more details, you can visit the [git-add documentation page on Git-scm](https://git-scm.com/docs/git-add)
* git commit: This command creates a new commit instance with the current content(s) of the index and with the log message describing the changes.  
  For more details, you can visit the [Git Commit](https://www.gitkraken.com/learn/git/commit) page in the GitKraken Learn Git center.
* git push: This command updates the remote refs with the local refs. In this example, we will update the master branch in our remote repository with our local changes.For more details, you can visit the [git-push documentation](https://www.google.com/url?q=https://git-scm.com/docs/git-push&sa=D&source=editors&ust=1626477349296000&usg=AOvVaw2A2gz1LCEWKdjOBjP-V1HT)[page on Git-scm](https://git-scm.com/docs/git-push)
* git pull: This command will pull all changes from a remote repository branch to your local repository branch.

7. Confirm the remote repository is in the GitHub account. Sure enough, the new file “firstCommit.txt” is there!

Now that you have received this Git Bash introduction, you can easily adopt these steps to your own projects as you integrate Git Bash with your GitHub account and use the Git version control system for your code base.

Alternatively, the process of integrating your command line terminal with Windows using GitKraken Client is much easier and takes far fewer steps. Simply download GitKraken Client on your Windows machine, open a new tab, and click **New Terminal Tab**. Then start typing your commands to get to work – seriously, it’s that easy!

## Using GitHub with GitKraken **Client**

In comparison to using GitHub with Git Bash, in GitKraken Client, you can leverage the full power of the robust GitHub integration by simply signing into GitKraken Client using your GitHub credentials. From there, it’s just a few short steps to [generate an SSH key and add to GitHub](https://www.gitkraken.com/learn/git/problems/github-add-ssh-key), and clone or fork a GitHub repository. Learn more about how developers [use GitKraken Client with GitHub](https://www.gitkraken.com/integrations/github) for a seamless Git workflow, including [creating GitHub pull requests](https://www.gitkraken.com/learn/git/problems/github-pull-requests).

## Set Git Bash as Your Default Terminal in GitKraken Client

Git Bash is a life saver for Windows users who want to leverage the power of the Git command line for their version control. It is easy to install Git Bash and start using it as stated throughout this article.

And even better, with [GitKraken Client](https://www.gitkraken.com/git-client), you can combine the power of Git Bash with visualization tools—like the commit graph, diff view, file history and blame, to unlock even more productivity.

To use Git Bash as the default shell for [GitKraken Client’s built in CLI](https://www.gitkraken.com/cli), perform the following steps:

* Select **Preferences**, represented by the gear icon, from the top toolbar
* From the left navigation, select **Terminal**
* Select **Default Terminal**
* From the subsequent dropdown menu, select **Git Bash**

You’re all set! Enjoy the power of Git Bash and the convenience of powerful visualization features, combined into one luxurious tool: GitKraken Client.

## Git Bash FAQ

### Q: How to Paste into Git Bash?

To copy text in Git Bash, hold **Shift** and use the left/right arrows to select the desired text, and hit **Enter**. To paste into Git Bash, press **Insert** on your keyboard.

### Q: How to Update Git Bash?

A: Updating Git Bash can be accomplished simply by updating Git for Windows. From a command line, run git update-git-for-windows.

# GitHub Cheat Sheet

Documentation Source: <https://education.github.com/git-cheat-sheet-education.pdf>

## SETUP

Configuring user information used across all local repositories

|  |
| --- |
| **git config --global user.name “[firstname lastname]”**  set a name that is identifiable for credit when review version history |
| **git config --global user.email “[valid-email]”**  set an email address that will be associated with each history marker |
| **git config --global color.ui auto**  set automatic command line coloring for Git for easy reviewing |

## SETUP & INIT

Configuring user information, initializing and cloning repositories

|  |
| --- |
| **git init**  initialize an existing directory as a Git repository |
| **git clone [url]**  retrieve an entire repository from a hosted location via URL |

## STAGE & SNAPSHOT

Working with snapshots and the Git staging area

|  |
| --- |
| **git status**  show modified files in working directory, staged for your next commit |
| **git add [file]**  add a file as it looks now to your next commit (stage) |
| **git reset [file]**  unstage a file while retaining the changes in working directory |
| **git diff**  diﬀ of what is changed but not staged |
| **git diff --staged**  diﬀ of what is staged but not yet committed |
| **git commit -m “[descriptive message]”**  commit your staged content as a new commit snapshot |

## BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

|  |
| --- |
| **git branch**  list your branches. a \* will appear next to the currently active branch |
| **git branch [branch-name]**  create a new branch at the current commit |
| **git checkout**  switch to another branch and check it out into your working directory |
| **git merge [branch]**  merge the specified branch’s history into the current one |
| **git log**  show all commits in the current branch’s history |

## INSPECT & COMPARE

Examining logs, diffs and object information

|  |
| --- |
| **git log**  show the commit history for the currently active branch |
| **git log branchB..branchA**  show the commits on branchA that are not on branchB |
| **git log --follow [file]**  show the commits that changed file, even across renames |
| **git diff branchB...branchA**  show the diﬀ of what is in branchA that is not in branchB |
| **git show [SHA]**  show any object in Git in human-readable format |

## TRACKING PATH CHANGES

Versioning file removes and path changes

|  |
| --- |
| **git rm [file]**  delete the file from project and stage the removal for commit |
| **git mv [existing-path] [new-path]**  change an existing file path and stage the move |
| **git log --stat -M**  show all commit logs with indication of any paths that moved |

## IGNORING PATTERNS

Preventing unintentional staging or committing of files

|  |
| --- |
| **logs/**  **\*.notes pattern\*/**  Save a file with desired patterns as .gitignore with either direct string matches or wildcard globs. |
| **git config --global core.excludesfile [file]**  system wide ignore pattern for all local repositories |

## SHARE & UPDATE

Retrieving updates from another repository and updating local repos

|  |
| --- |
| **git remote add [alias] [url]**  add a git URL as an alias |
| **git fetch [alias]**  fetch down all the branches from that Git remote |
| **git merge [alias]/[branch]**  merge a remote branch into your current branch to bring it up to date |
| **git push [alias] [branch]**  Transmit local branch commits to the remote repository branch |
| **git pull**  fetch and merge any commits from the tracking remote branch |

## REWRITE HISTORY

Rewriting branches, updating commits and clearing history

|  |
| --- |
| **git rebase [branch]**  apply any commits of current branch ahead of specified one |
| **git reset --hard [commit]** |
| clear staging area, rewrite working tree from specified commit |

## TEMPORARY COMMITS

Temporarily store modified, tracked files in order to change branches

|  |
| --- |
| **git stash**  Save modified and staged changes |
| **git stash list**  list stack-order of stashed file changes |
| **git stash pop**  write working from top of stash stack |
| **git stash drop**  discard the changes from top of stash stack |

# GitHub Learning Lab

Documentation Source: <https://github.com/apps/github-learning-lab>

## GitHub App

Level up your GitHub skills with GitHub Learning Lab. Our friendly bot will take you through a series of fun, practical projects that will give you the skills you need in no time--and share helpful feedback along the way.

* **Learn the way you work:** Work directly in the GitHub repositories, issues, and pull requests with guidance and feedback from our bot
* **Pick up new skills:** Discover new challenges and learn how to work better with GitHub
* **Track your accomplishments:** See the courses you’ve completed on your profile page

## Getting started with GitHub Learning Lab

Install GitHub Learning Lab as you would install any GitHub App. GitHub Apps take extra steps to be respectful of your privacy but require a little bit of setup to get started.

GitHub Learning Lab creates a repository with your GitHub user account every time you register for a new course. This repository is where you'll work with Learning Lab in issues and pull requests.

When you install the app, you can choose to authorize access to all repositories, or grant it access to a single repository.

### Recommended approach: Install GitHub Learning Lab on all repositories

* Give GitHub Learning Lab general access and trust at your user account level, just like you would for any other OAuth service. This way, you’ll only need to authorize GitHub Learning Lab once on initial installation.

### Alternative approach: Install GitHub Learning Lab on a single repository

* Grant GitHub Learning Lab access to one repository on your account. Once this permission is granted, the app will be able to create new repositories on your account.
* As you register for each course, the app will create a new repository. To work with the bot on a specific course, you’ll need to grant GitHub Learning Lab access to that repository.