# 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

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

To sign up for an account on GitHub.com, navigate to 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."

### 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."

### 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."

### 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."

Optionally, after you have configured 2FA, add a passkey to your account to enable a secure, passwordless login. For more information, see "About passkeys" and "Managing your passkeys."

### 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" and "Viewing contributions on your profile."

## Part 2: Using GitHub's 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

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" guide. For more information about how Git is used on GitHub, see "GitHub flow."

## 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."

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."

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."

## 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."

## 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."

You can learn GitHub Flavored Markdown with the "Communicate using Markdown" course on GitHub Skills.

## 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."

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."

### 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."

## 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

#### 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."

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."

#### 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."

#### 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."

## 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."

## 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."

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."

## 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."

## 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" 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."

## 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" and "About Projects."

## 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."

### 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."

### 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."

## 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

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.

### 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, or create integrations that extend GitHub. For more information, see "Comparing GitHub's REST API and GraphQL API."

### 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."

### 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."

## 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."

### 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."

### 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."

## Part 6: Participating in GitHub's 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

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" 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. For more information, see "Finding ways to contribute to open source on GitHub."

## 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."

## 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."

## 4. Learning with GitHub Skills

You can learn new skills by completing fun, realistic projects in your very own GitHub repository with GitHub Skills. 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."

## 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."

## 6. Contacting GitHub Support

GitHub Support can help you troubleshoot issues you run into while using GitHub. For more information, see "About GitHub Support."