GitHub 101

Source:

- GitHub Docs, GitHub, https://docs.github.com/en/get-started/start-your-journey Accessed 01/19/2026.
- GitHub Pages documentation, GitHub, https://docs.github.com/en/pages Accessed 01/19/2026.

About GitHub and Git

You can use GitHub and Git to collaborate on work.

About GitHub

GitHub is a cloud-based platform where you can store, share, and work together with others to write code.

Storing your code in a "repository" on GitHub allows you to:

- **Showcase or share** your work.
- **Track and manage** changes to your code over time.
- Let others **review** your code, and make suggestions to improve it.
- **Collaborate** on a shared project, without worrying that your changes will impact the work of your collaborators before you're ready to integrate them.

Collaborative working, one of GitHub's fundamental features, is made possible by the open-source software, Git, upon which GitHub is built.

About Git

Git is a version control system that intelligently tracks changes in files. Git is particularly useful when you and a group of people are all making changes to the same files at the same time.

Typically, to do this in a Git-based workflow, you would:

- **Create a branch** off from the main copy of files that you (and your collaborators) are working on.
- **Make edits** to the files independently and safely on your own personal branch.

- Let Git intelligently **merge** your specific changes back into the main copy of files, so that your changes don't impact other people's updates.
- Let Git **keep track** of your and other people's changes, so you all stay working on the most up-to-date version of the project.

To try using Git yourself, see Getting started with Git.

How do Git and GitHub work together?

When you upload files to GitHub, you'll store them in a "Git repository." This means that when you make changes (or "commits") to your files in GitHub, Git will automatically start to track and manage your changes.

There are plenty of Git-related actions that you can complete on GitHub directly in your browser, such as creating a Git repository, creating branches, and uploading and editing files.

However, most people work on their files locally (on their own computer), then continually sync these local changes—and all the related Git data—with the central "remote" repository on GitHub. There are plenty of tools that you can use to do this, such as GitHub Desktop.

Once you start to collaborate with others and all need to work on the same repository at the same time, you'll continually:

- **Pull** all the latest changes made by your collaborators from the remote repository on GitHub.
- **Push** back your own changes to the same remote repository on GitHub.

Git figures out how to intelligently merge this flow of changes, and GitHub helps you manage the flow through features such as "pull requests."

Where do I start?

If you're new to GitHub, and unfamiliar with Git, we recommend working through the articles in the Start your journey category. The articles focus on tasks you can complete directly in your browser on GitHub and will help you to:

- **Create an account** on GitHub.
- **Learn the "GitHub Flow"**, and the key principles of collaborative working (branches, commits, pull requests, merges).

- **Personalise your profile** to share your interests and skills.
- **Explore GitHub** to find inspiration for your own projects and connect with others.
- Learn how to **download** interesting code for your own use.
- Learn how to **upload** something you're working on to a GitHub repository.

Creating an account on GitHub

Create a personal account to get started with GitHub.

About your personal account on GitHub

To get started with GitHub, you'll need to create a free personal account and verify your email address.

You can also authenticate with Google or Apple - which are the supported social login providers when you create your account on GitHub. For iOS users, even if you have enabled the setting "Hide My Email addresses" for your Apple account, using social login will result in creating a new GitHub account.

Every person who uses GitHub signs in to a user account. Your user account is your identity on GitHub and has a username and profile. For example, see @octocat's profile.

Later, you can explore the different types of accounts that GitHub offers, and decide if you need a billing plan. For more information, see Types of GitHub accounts and GitHub's plans.

Note that the steps in this article don't apply to Enterprise Managed Users. If your GitHub account has been created for you by your company, you can skip this article and continue to Hello World.

Signing up for a new personal account

1. Navigate to https://github.com/.
2. Click **Sign up**.
3. Alternatively, click on **Continue with Google** to sign up using social login.
4. Follow the prompts to create your personal account.

During sign up, you'll be asked to verify your email address. Without a verified email address, you won't be able to complete some basic GitHub tasks, such as creating a repository.

Some enterprises create managed user accounts for their users. You can't sign up for a personal account with an email address that's already verified for a managed user account.

If you're having problems verifying your email address, there are some troubleshooting steps you can take. For more information, see Verifying your email address.

What is GitHub Pages?

You can use GitHub Pages to host a website about yourself, your organization, or your project directly from a repository on GitHub.

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.

About GitHub Pages

GitHub Pages is a static site hosting service that takes HTML, CSS, and JavaScript files straight from a repository on GitHub, optionally runs the files through a build process, and publishes a website. You can see examples of GitHub Pages sites in the GitHub Pages examples collection.

Types of GitHub Pages sites

There are two types of GitHub Pages sites. Sites associated with a user or organization account, and sites for a specific project.

Property	User and organization sites	Project sites
Source files	Must be stored in a repository named `<owner>.github.io`, where `<owner>` is the personal or organization account name	Stored in a folder within the repository that contains the project's code
Limits	Maximum of one pages site per account	Maximum of one pages site per repository
Default site location	`http(s)://<owner>.github.io`	`http(s)://<owner>.github.io/<repositoryname>`

Hosting on your own custom domain

You can host your site on GitHub's `github.io` domain or your own custom domain. See Configuring a custom domain for your GitHub Pages site.

Data collection

When a GitHub Pages site is visited, the visitor's IP address is logged and stored for security purposes, regardless of whether the visitor has signed into GitHub or not. For more information about GitHub's security practices, see GitHub Privacy Statement.

Creating a GitHub Pages site

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

Creating a repository for your site

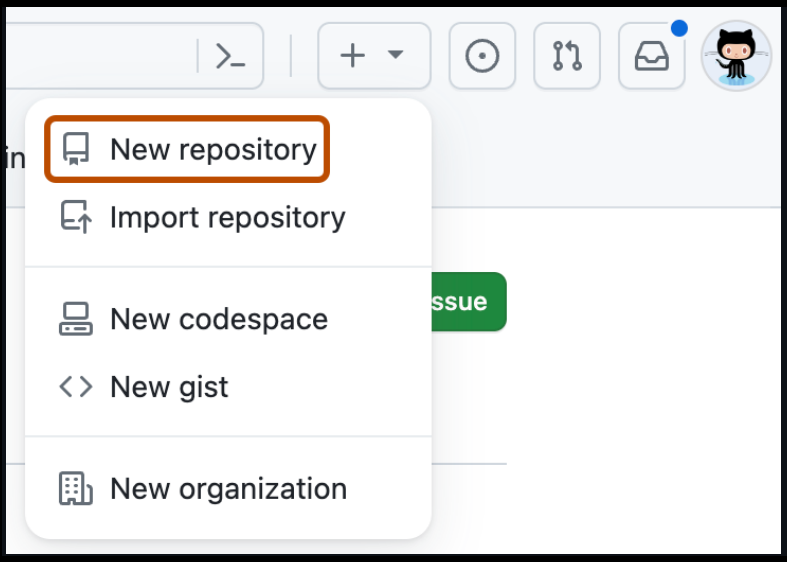
You can either create a repository or choose an existing repository for your site.

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.

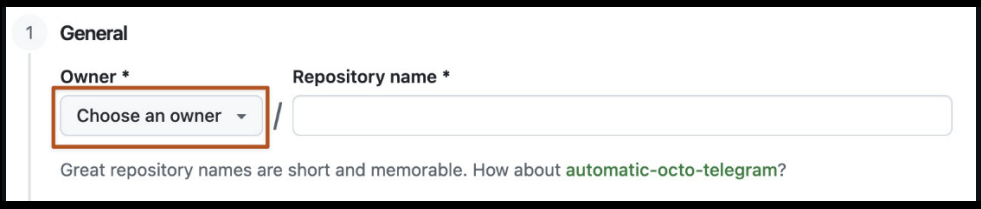
If the account that owns the repository uses GitHub Free or GitHub Free for organizations, the repository must be public.

If you want to create a site in an existing repository, skip to the Creating your site section.

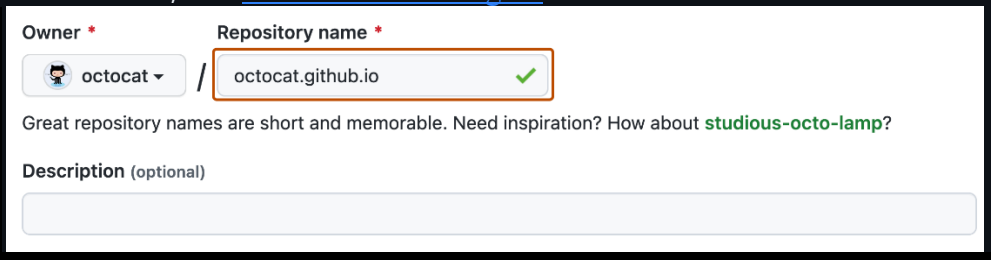
1. In the upper-right corner of any page, select, then click **New repository**.



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



3. 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 What is GitHub Pages?.



4. Choose a repository visibility. For more information, see About repositories.

5. Toggle **Add README** to **On**.
6. Click **Create repository**.

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.

Warning

GitHub Pages sites are publicly available on the internet, even if the repository for the site is private (if your plan or organization allows it). 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.

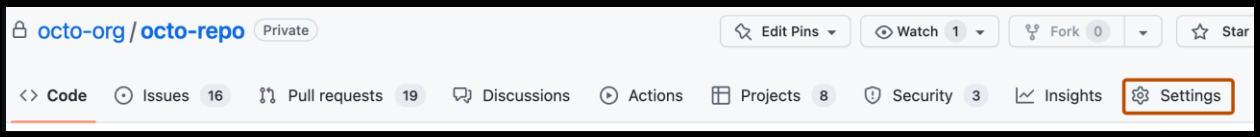
1. On GitHub, navigate to your site's repository.
2. Decide which publishing source you want to use. See Configuring a publishing source for your GitHub Pages site.
3. Create the entry file for your site. GitHub Pages will look for an `index.html`, `index.md`, or `README.md` file as the entry file for 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.
4. Configure your publishing source. See Configuring a publishing source for your GitHub Pages site.
5. Your GitHub Pages site is built and deployed with a GitHub Actions workflow. For more information, see 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 Billing and usage.

Viewing your published site

1. Under your repository name, click **Settings**. If you cannot see the "Settings" tab, select the dropdown menu, then click **Settings**.



2. In the "Code and automation" section of the sidebar, click **Pages**.
3. 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.

- 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.

Static site generators

GitHub Pages publishes any static files that you push to your repository. You can create your own static files or use a static site generator to build your site for you. You can also customize your own build process locally or on another server.

If you use a custom build process or a static site generator other than Jekyll, you can write a GitHub Actions workflow to build and publish your site. GitHub provides workflow templates for several static site generators. For more information, see Configuring a publishing source for your GitHub Pages site.

If you publish your site from a source branch, GitHub Pages will use Jekyll to build your site by default. If you want to use a static site generator other than Jekyll, we recommend that you write a GitHub Actions to build and publish your site instead. Otherwise, disable the Jekyll build process by creating an empty file called `.nojekyll` in the root of your publishing source, then follow your static site generator's instructions to build your site locally.

Note

GitHub Pages does not support server-side languages such as PHP, Ruby, or Python.

MIME types on GitHub Pages

A MIME type is a header that a server sends to a browser, providing information about the nature and format of the files the browser requested. GitHub Pages supports more than 750 MIME types across thousands of file extensions. The list of supported MIME types is generated from the mime-db project.

While you can't specify custom MIME types on a per-file or per-repository basis, you can add or modify MIME types for use on GitHub Pages. For more information, see the mime-db contributing guidelines.

GitHub Pages limits

Learn about the limits and limitations of GitHub Pages.

Usage limits

GitHub Pages is not intended for or allowed to be used as a free web-hosting service to run your online business, e-commerce site, or any other website that is primarily directed at either facilitating commercial transactions or providing commercial software as a service (SaaS). GitHub Pages sites shouldn't be used for sensitive transactions like sending passwords or credit card numbers.

In addition, your use of GitHub Pages is subject to the GitHub Terms of Service, including the restrictions on get-rich-quick schemes, sexually obscene content, and violent or threatening content or activity.

GitHub Pages sites are subject to the following usage limits:

- You can only create one user or organization site for each account on GitHub.
- GitHub Pages source repositories have a recommended limit of 1 GB. For more information, see About large files on GitHub.

- Published GitHub Pages sites may be no larger than 1 GB.
- GitHub Pages deployments will timeout if they take longer than 10 minutes.
- GitHub Pages sites have a *soft* bandwidth limit of 100 GB per month.
- GitHub Pages sites have a *soft* limit of 10 builds per hour. This limit does not apply if you build and publish your site with a custom GitHub Actions workflow.
- In order to provide consistent quality of service for all GitHub Pages sites, rate limits may apply. These rate limits are not intended to interfere with legitimate uses of GitHub Pages. If your request triggers rate limiting, you will receive an appropriate response with an HTTP status code of 429, along with an informative HTML body.

If your site exceeds these usage quotas, we may not be able to serve your site, or you may receive a polite email from GitHub Support suggesting strategies for reducing your site's impact on our servers, including putting a third-party content distribution network (CDN) in front of your site, making use of other GitHub features such as releases, or moving to a different hosting service that might better fit your needs.

Educational exercises

Using GitHub Pages to create a copy of an existing website as a learning exercise is not prohibited. However, in addition to complying with the GitHub Terms of Service, you must write the code yourself, the site must not collect any user data, and you must include a prominent disclaimer on the site indicating that the project is not associated with the original and was only created for educational purposes.