

GitHub

Everything you want to know

What is GitHub?

GitHub is a web-based version control and collaboration platform for software developers. Microsoft, the biggest single contributor to GitHub, acquired the platform for \$7.5 billion in 2018. GitHub, which is delivered through a software as a service ([SaaS](#)) business model, was started in 2008. It was founded on Git, an open source [code](#) management system created by Linus Torvalds to make software builds faster.

Git is used to [store the source code for a project](#) and track the complete history of all changes to that code. It lets developers collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers.

GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans. Each public and private repository contains all of a project's files, as well as each file's revision history. Repositories can have multiple collaborators and can be either public or private.

How does GitHub work?

GitHub facilitates social coding by providing a hosting service and web interface for the Git code repository, as well as management tools for collaboration. The developer platform can be thought of as a [social networking](#) site for software developers. Members can follow each other, rate each other's work, receive updates for specific open source projects, and communicate publicly or privately. The following are some important terms GitHub developers use:

- **Fork.** A [fork](#), also known as a branch, is a repository that has been copied from one member's account to another member's account. Forks and branches let a developer make modifications without affecting the original code.
- **Pull request.** If a developer would like to share their modifications, they can send a pull request to the owner of the original repository.
- **Merge.** If, after reviewing the modifications, the original owner would like to pull the modifications into the repository, they can accept the modifications and merge them with the original repository.
- **Push.** This is the reverse of a pull -- a programmer sends code from a local copy to the online repository.
- **Commit.** A [commit, or code revision](#), is an individual change to a file or set of files. By default, commits are retained and [interleaved onto the main project](#), or they can be combined into a simpler merge via commit squashing. A unique ID is created when each commit is saved that lets collaborators keep a record of their work. A commit can be thought of as a snapshot of a repository.
- **Clone.** A clone is a local copy of a repository.

Benefits and features of GitHub

GitHub facilitates collaboration among developers. It also provides distributed version control. Teams of developers can work together in a centralized Git repository and track changes as they go to stay organized.

GitHub offers an on-premises version in addition to the well-known SaaS product. GitHub Enterprise supports integrated development environments and continuous integration tools, as well as many third-party apps and services. It offers more security and auditability than the SaaS version.

Other products and features of note include the following:

- **GitHub Gist** lets users share pieces of code or other notes.
- **GitHub Flow** is a lightweight, branch-based workflow for regularly updated deployments.
- **GitHub Pages** are static webpages to [host a project](#), pulling information directly from an individual's or organization's GitHub repository.
- **GitHub Desktop** lets users access GitHub from Windows or Mac desktops, rather than going to GitHub's website.
- **GitHub Student Developer Pack** is a free offering of developer tools for students. It includes cloud resources, programming tools and support, and GitHub access.
- **GitHub Campus Experts** is a program students can use to become leaders at their schools and develop technical communities there.
- **GitHub CLI** is a free, open source [command-line](#) tool that brings GitHub features, such as pull requests, to a user's local terminal. This capability eliminates the need to switch contexts when coding, streamlining workflows.
- **GitHub Codespaces** is a cloud-based development environment that gives users access to [common programming languages](#) and tools. The coding environment runs in a [container](#) and gives users a certain amount of free time before switching to a paid pricing model.

GitHub use cases

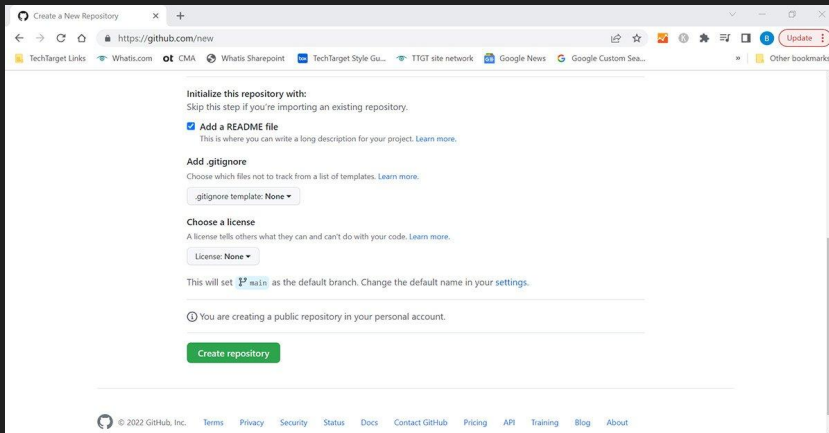
GitHub is used to store, track and collaborate on software projects in a number of different contexts:

- **Businesses** use GitHub as version control systems, letting development team members [track changes to source code](#) as developers collaborate on it. This lets different coders work on a project simultaneously and ensures everyone is working on the latest version of the code, simplifying project management. It also allows for previous versions to be called upon should developers need to reference them. GitHub enables code sharing among developers because code is stored in a central location. GitHub Enterprise also helps with regulatory compliance because it is a standardized way to store code.
- **Programming instructors and students** make use of GitHub in several ways. The Student Developer Pack gives teachers and students an array of low-cost resources. Students use the platform to learn web development, work on creative development projects and host virtual events.
- **Open source software developers** use GitHub to share projects with individuals who want to use their software or collaborate on it. Developers network, collaborate and pitch their work to other developers in real time, catching errors in proposed code before changes are finalized. These collaboration and networking capabilities are why GitHub is classified as a [social media](#) site; it often links to other community sites such as [Reddit](#) in the repository notes. Users also can download applications from GitHub.
- **Nonprogrammers** also use GitHub to work on document-based and multimedia projects. The platform is intuitive to use, and its version control tools are useful for collaboration. For example, *The Art of the Command Line* is a [comprehensive guide](#) to the command line. Samplebrain is an experimental [music production tool](#) by electronic musician Aphex Twin. And the *Open Source Cookbook* is a collection of [food recipes](#).

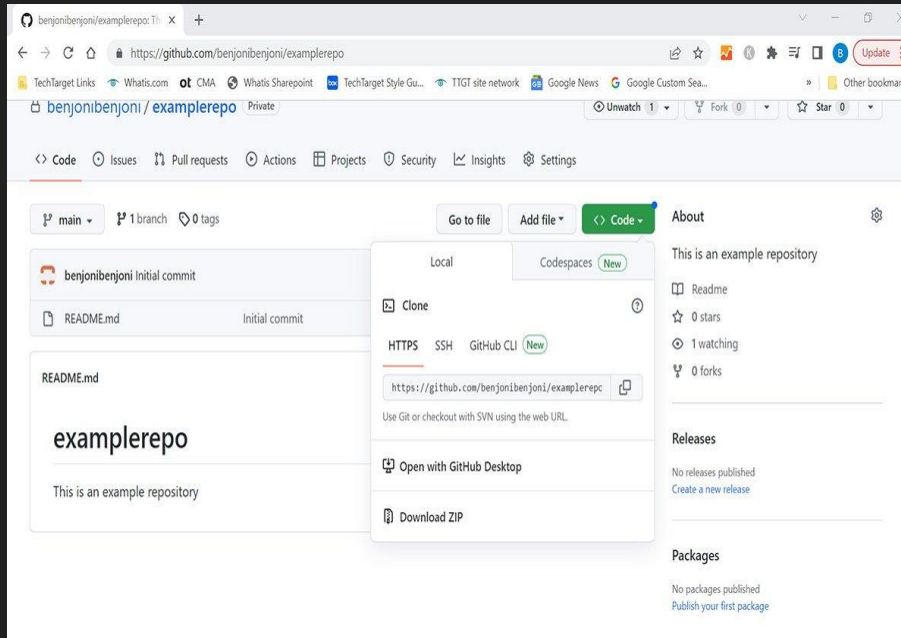
Getting started on GitHub

To sign up for GitHub and create a repository, new users and beginners follow these steps:

- **Learn about the command line.** The command line is how users interact with GitHub. The ability to use it is a [prerequisite for working with GitHub](#); tutorials and other tools are available to help with this process. An alternative is the GitHub Desktop client.
- **Install Git.** Git can be installed for free using instructions on the Git website. Installing GitHub Desktop will also install a command-line version of Git. Git comes installed by default on many Mac and Linux machines.
- **Create an account.** Go to GitHub's website and create a GitHub account using an email address.
- **Create a new repository.** Go to the GitHub homepage, click the + sign and then click **examplerpo**. Name the repository and provide a brief description when prompted. Add a README file, .gitignore template and project license. Then scroll to the bottom of the page and click **Create repository**.



The user should now have a blank repository on their GitHub page. They can [create a local copy](#) of that repository using the `git init` command in the terminal.



once a repository has been created in GitHub, users can clone a local copy.