



GitHub Glossary

Below are a list of some Git and GitHub specific terms we use across our sites and documentation.

If you see some Git-related terminology not listed here, you might find an explanation of it in [Git Reference](#) or our [Git SCM](#) book.

Blame

The "blame" feature in Git describes the last modification to each line of a file, which generally displays the revision, author and time. This is helpful, for example, in tracking down when a feature was added, or which commit led to a particular bug.

Branch

A branch is a parallel version of a repository. It is contained within the repository, but does not affect the primary or `master` branch allowing you to work freely without disrupting the "live" version. When you've made the changes you want to make, you can merge your branch back into the `master` branch to publish your changes.

Clone

A clone is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. It is, however, connected to the remote version so that changes can be synced between the two. You can push your local changes to the [remote](#) to keep them synced when you're online.

Collaborator

A collaborator is a person with read and write access to a repository who has been [invited to contribute](#) by the repository owner.

Commit

A commit, or "revision", is an individual change to a file (or set of files). It's like when you save a file, except with Git, every time you save it creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by who. Commits usually contain a commit message which is a brief description of what changes were made.

Contributor

A contributor is someone who has contributed to a project by having a pull request merged but does not have collaborator access.

Diff

A diff is the *difference* in changes between two commits, or saved changes. The diff will visually describe what was added or removed from a file since its last commit.

Fetch

Fetching refers to getting the latest changes from an online repository (like GitHub.com) without merging

Article versions

[GitHub.com](#)

[GitHub Enterprise 2.3](#)

[GitHub Enterprise 2.2](#)

[GitHub Enterprise 2.1](#)

[GitHub Enterprise 2.0](#)

them in. Once these changes are fetched you can compare them to your local branches (the code residing on your local machine).

Fork

A fork is a personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original. Forks remain attached to the original, allowing you to submit a pull request to the original's author to update with your changes. You can also keep your fork up to date by pulling in updates from the original.

Git

Git is an open source program for tracking changes in text files. It was written by the author of the Linux operating system, and is the core technology that GitHub, the social and user interface, is built on top of.

Issue

Issues are suggested improvements, tasks or questions related to the repository. Issues can be created by anyone (for public repositories), and are moderated by repository collaborators. Each issue contains its own discussion forum, can be labeled and assigned to a user.

Markdown

Markdown is an incredibly simple semantic file format, not too dissimilar from .doc, .rtf and .txt. Markdown makes it easy for even those without a web-publishing background to write prose (including with links, lists, bullets, etc.) and have it displayed like a website. GitHub supports Markdown, and you can learn about the semantics [here](#).

Merge

Merging takes the changes from one branch (in the same repository or from a fork), and applies them into another. This often happens as a Pull Request (which can be thought of as a request to merge), or via the command line. A merge can be done automatically via a Pull Request via the GitHub.com web interface if there are no conflicting changes, or can always be done via the command line. See [Merging a pull request](#).

Open Source

Open source software is software that can be [freely used, modified, and shared \(in both modified and unmodified form\) by anyone](#). Today the concept of "open source" is often extended beyond software, to represent a philosophy of collaboration in which working materials are made available online for anyone to fork, modify, discuss, and contribute to.

Organizations

Organizations are a group of two or more users that typically mirror real-world organizations. They are administered by users and can contain both repositories and teams.

Private Repository

Private repositories are repositories that can only be viewed or contributed to by their creator and collaborators the creator specified.

Pull

Pull refers to when you are fetching *in* changes *and* merging them. For instance, if someone has edited the remote file you're both working on, you'll want to *pull* in those changes to your local copy so that it's up to date.

Pull Request

Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators. Like issues, pull requests each have their own discussion forum. See [Using Pull Requests](#).

Push

Pushing refers to sending your committed changes to a remote repository such as GitHub.com. For instance, if you change something locally, you'd want to then *push* those changes so that others may access them.

Remote

This is the version of something that is hosted on a server, most likely GitHub.com. It can be connected to local clones so that changes can be synced.

Repository

A repository is the most basic element of GitHub. They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.

SSH Key

SSH keys are a way to identify yourself to an online server, using an encrypted message. It's as if your computer has its own unique password to another service. GitHub uses SSH keys to securely transfer information from GitHub.com to your computer.

Upstream

When talking about a branch or a fork, the primary branch on the original repository is often referred to as the "upstream", since that is the main place that other changes will come in from. The branch/fork you are working on is then called the "downstream".

User

Users are personal GitHub accounts. Each user has a personal profile, and can own multiple repositories, public or private. They can create or be invited to join organizations or collaborate on another user's repository.

See Also

› [The Official Git Glossary](#)

 **Contact a human**

