



# git terminology

## 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.

## Fetch

Fetching refers to getting the latest changes from an online repository without merging them in. Once these changes are fetched you can compare them to your local branches (the code residing on your local machine).

## Push

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

## Clone

A clone is 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.

## 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.

## HEAD

Git's way of referring to the current snapshot. Internally, the git checkout command simply updates the HEAD to point to either the specified branch or commit. When it points to a branch, Git doesn't complain, but when you check out a commit, it switches into a "detached HEAD" state.

## Detached HEAD State

The HEAD pointer in Git determines your current working revision (and thereby the files that are placed in your project's working directory). Normally, when checking out a proper branch name, Git automatically moves the HEAD pointer along when you create a new commit. You are automatically on the newest commit of the chosen branch.

When you instead choose to check out a commit hash, Git won't do this for you. The consequence is that when you make changes and commit them, these changes do NOT belong to any branch.

This means they can easily get lost once you check out a different revision or branch: not being recorded in the context of a branch, you lack the possibility to access that state easily (unless you have a brilliant memory and can remember the commit hash of that new commit...).

## Origin

The default name of the remote that was used to clone the repository from.

## 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.

## Fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

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.

## 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.

## Git Bash

Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell. First released in 1989, it has been distributed widely as it is a default shell on the major Linux distributions and OS X.

Git Bash for Windows is not just bash compiled for Windows. It's package that contains bash and a collection of separate \*nix utilities like ssh, scp, cat, find and others.

## Staged files

To stage a file is to prepare it for a commit. Because git exposes this action to the users control it allows you to create partial commits, or to modify a file, stage it, modify it again, and only commit or revert to the original modification.

Staging allows you finer control over exactly how you want to approach version control.

## Master

The default development branch. Whenever you create a git repository, a branch named "master" is created, and becomes the active branch.

## 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.

## Index, Staging area

This is a 'cache' between your working directory and your repository. You can add changes to the index and build your next commit step by step. When your index content is to your likes you can create a commit from it. Also used to keep information during failed merges (your side, their side and current state)

## Tracked files

Tracked files are files that were in the last snapshot; they can be unmodified, modified, or staged. Untracked files are everything else – any files in your working directory that were not in your last snapshot and are not in your staging area. When you first clone a repository, all of your files will be tracked and unmodified because Git just checked them out and you haven't edited anything.

## 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.

## Rebase

Rebasing a branch or more generally a series of commits means moving these commits from one base to another where “base” is the predecessor commit of the oldest commit in the series. Rebasing is often used as an alternative to merging. The actual branch that is being rebased is removed and the commits from that branch are applied on top of the commits in the merge target branch. This technique is used to keep “linear” commit history.

## Stash

Git allows you to "stash away" changes. This gives you a clean working tree without any changes. Later they can be "popped" to be brought back. This can be a life saver if you need to temporarily work on an unrelated change (eg. time critical bug fix)

## Hook

A script that runs automatically every time a particular event occurs in a Git repository. Hooks let you customize Git’s internal behaviour and trigger customizable actions at key points in the development life cycle.

## Remote

This is the version of the repo that is hosted somewhere else. It can be connected to local clones so that changes can be synced.

## 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".

## Tag

A reference typically used to mark a particular point in the commit chain. In contrast to a head, a tag is not updated by the commit command.

## Pull Request

Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators.



[goo.gl/VaDaHw](https://goo.gl/VaDaHw)