Short Git and GitHub Glossary

Official Glossary

The short glossary below was condensed and modified from the glossary in the official GitHub docs. The full glossary is at github-glossary.

branch

A branch is a parallel version of a repository. It is contained within the repository, but does not affect the primary or main 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 main branch to publish your changes.

checkout

You can use git checkout on the command line to create a new branch, change your current working branch to a different branch, or even to switch to a different version of a file from a different branch with git checkout [branchname] [path to file]. The "checkout" action updates all or part of the working tree with a tree object or blob from the object database, and updates the index and HEAD if the whole working tree is pointing to a new branch.

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. When you make a clone, you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. The repository you cloned is still connected to the remote version so that you can push your local changes to the remote to keep them synced when you're online.

commit

A commit, or "revision", is an individual change to a file (or set of files). When you make a commit to save your work, Git creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of the specific changes committed along with who made them and when. Commits usually contain a commit message which is a brief description of what changes were made.

commit message

Short, descriptive text that accompanies a commit and communicates the change the commit is introducing.

default branch

The base branch for new pull requests and code commits in a repository. Each repository has at least one branch, which Git creates when you initialize the repository. The first branch is usually called {% ifversion ghes < 3.2 %}master{% else %}main{% endif %}, and is often the default branch.

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

When you use git fetch, you're adding changes from the remote repository to your local working branch without committing them. Unlike git pull, fetching allows you to review changes before committing them to your local branch.

force push

A Git push that overwrites the remote repository with local changes without regard for conflicts.

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 upstream repository. You can also open a pull request in the upstream repository and keep your fork synced with the latest changes since both repositories are still connected.

main

The default development branch. Whenever you create a Git repository, a branch named main is created, and becomes the active branch. In most cases, this contains the local development, though that is purely by convention and is not required.

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 through a pull request via the GitHub.com web interface if there are no conflicting changes, or can always be done via the command line.

merge conflict

A difference that occurs between merged branches. Merge conflicts happen when people make different changes to the same line of the same file, or when one person edits a file and another person deletes the same file. The merge conflict must be resolved before you can merge the branches.

origin

The default upstream repository. Most projects have at least one upstream project that they track. By default, origin is used for that purpose.

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. See also fetch.

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.

push

To push means to send your committed changes to a remote repository on GitHub.com. For instance, if you change something locally, you can push those changes so that others may access them.

README

A text file containing information about the files in a repository that is typically the first file a visitor to your repository will see. A README file, along with a repository license, contribution guidelines, and a code of conduct, helps you share expectations and manage contributions to your project.

remote

This is the version of a repository or branch that is hosted on a server, most likely GitHub.com. Remote versions 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.

resolve

The action of fixing up manually what a failed automatic merge left behind.

review

Reviews allow others with access to your repository to comment on the changes proposed in pull requests, approve the changes, or request further changes before the pull request is merged.

staged

A new file is "staged" if it is ready to be included in the repo on the next commit. Done with `git add`.

status

A visual representation within a pull request that your commits meet the conditions set for the repository you're contributing to. Obtained with `git status`

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". Also called origin.