

Git Glossary

Informal Definition

Formal Definition

git

Git is an open source program to keep track of changes in your files. It just takes screen shots of your files when you save them, and then it compares the files for differences.

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.

branch

A branch is a mirror of your current repo. When you branch, your current repo (usually your main branch) goes untouched. This is great when you want to create a feature in your application, because you do not have to worry about breaking a working application since you are branched off of it.

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 when you “clone” a remote repository and receive a local copy for your own modification. Git will keep track of all your modifications locally, so you don't have to depend on the reliability of the server.

note: if you commit changes locally, it will not effect the files on the remote repository. If you want your files to be in sync with the files on the remote repository, you will want you do a push.

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.

push

A push is how you get your local files to a remote repo so your team will have the option to pull your latest modifications.

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.

merge

Merge is a definition within itself. If you were working on a branch in which you have your working feature built, and you want to merge that feature back to the master branch, you will need to do a merge. It combines them into one.

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.

pull

A pull gets the latest code from a remote repository. When you do a pull, it also does a merge. A pull is basically a fetch and a merge.

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.

fetch

A fetch is a pull without the merge. This is helpful if you want to compare the code your team has put together before deciding to merge it to your local repository.

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

checkout

Checkout lets you navigate between branches.

The git checkout command lets you navigate between the branches created by git branch. Checking out a branch updates the files in the working directory to match the version stored in that branch, and it tells Git to record all new commits on that branch.