1. Intro

Time to level up again! Version control super powers!

Git tag = add tags to specific commits. Extra label for a commit that can indicate useful information.

Git branch = allows multiple lines of development.

Git checkout = switch between different branches.

Git merge = combines changes on different branches.

1. Tagging

A tag stays locked to a commit even when more commits are added to a repository.

git tag -a v1.0 -m "Add tag"

git log –decorate is a default in git 2.13

$ git tag -d v1.0

Deleted tag 'v1.0' (was 996e6a5)

Adding A Tag To A Past Commit

Running git tag -a v1.0 will tag the most recent commit. But what if you wanted to tag a commit that occurred farther back in the repo's history?

All you have to do is provide the SHA of the commit you want to tag!

$ git tag -a v1.0 a87984

1. Branching

Master is the default branch.

Git checkout sidebar modes the head from master to sidebar.

The git branch command is used to interact with Git's branches:

$ git branch

It is important to switch to the new branch after creating it.

$ git branch sidebar

$ git checkout sidebar

Created and moved to sidebar branch.

$ git branch -d sidebar

Deletes the sidebar branch.

1. Branching Effectively

Create and switch in one command!

$ git checkout -b footer master

Switched to a new branch 'footer'

See All Branches At Once

git log --oneline --decorate --graph --all

1. Merging

Making a merge makes a commit.

To undo a merge on the wrong branch, use git reset --hard HEAD^

The git merge command is used to combine Git branches:

$ git merge <name-of-branch-to-merge-in>

1. Merge Conflicts

Choose what to keep, save files, add them, recommit.

1. Outro