

Git Basics

Sabari Kumar

CSU Theory Group Coding Camp

Summer 2022



Colorado State University

The Need for Version Control

- Case 1: The normal code development cycle: oops, I broke something!
- Case 2: I have collaborators that work on different parts of the code
- Case 3: I want to sync my code across computers

Git vs. Github



- Software package released in 2005
- Runs locally
- Stores snapshots of a project

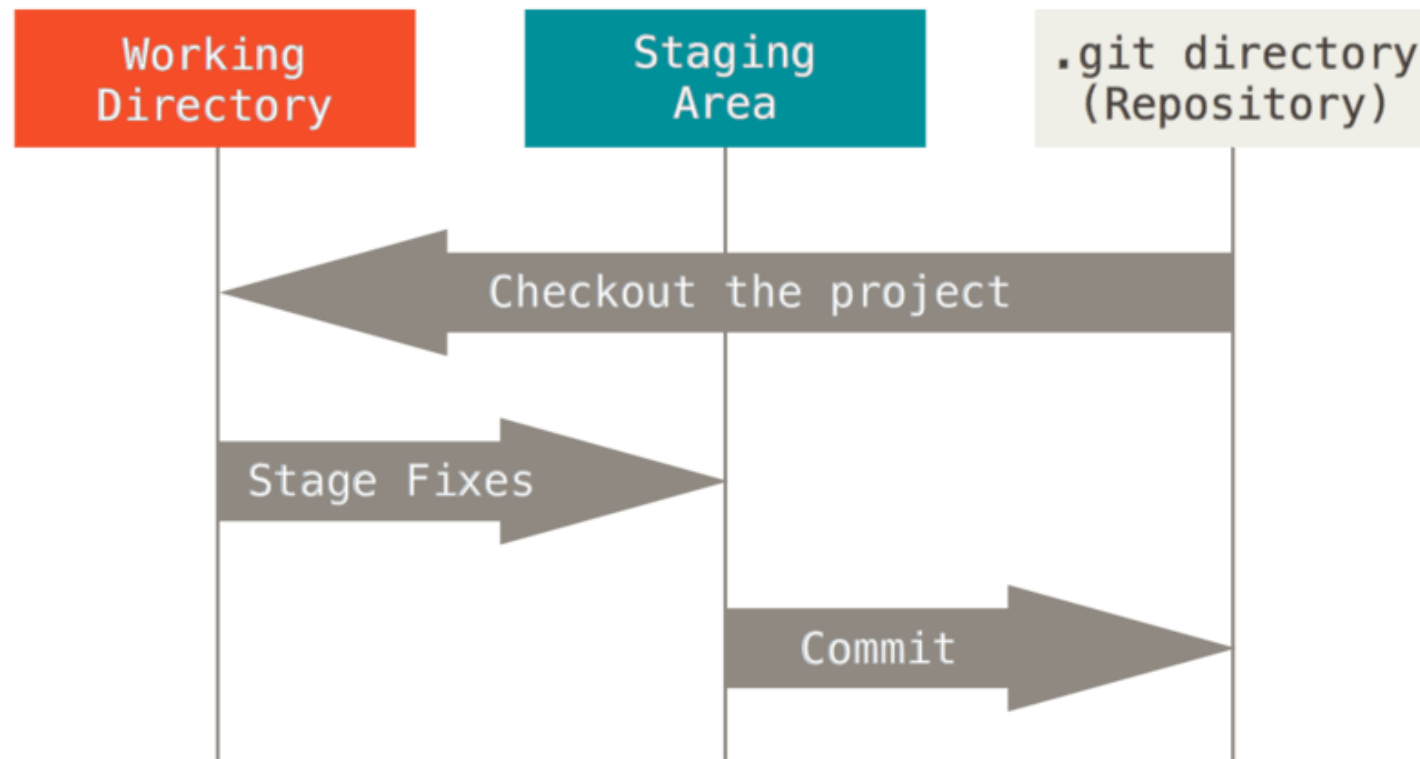


- Commercial company
- Hosts files on the internet

More on Git

- Operations are local (until you connect to github)
- Git uses snapshots, as opposed to diffs
- Hashes of files are used for
labels: 24b9da6552252987aa493b52f8696cd6d3b00373

Basic Git Workflow:



Basic Git Workflow Ctd.

- Checkout your project:
 - `git init` for local files
 - `git clone` for remote repositories
- Make your changes
- Stage changes for commit – `git add`
- Commit - `git commit`

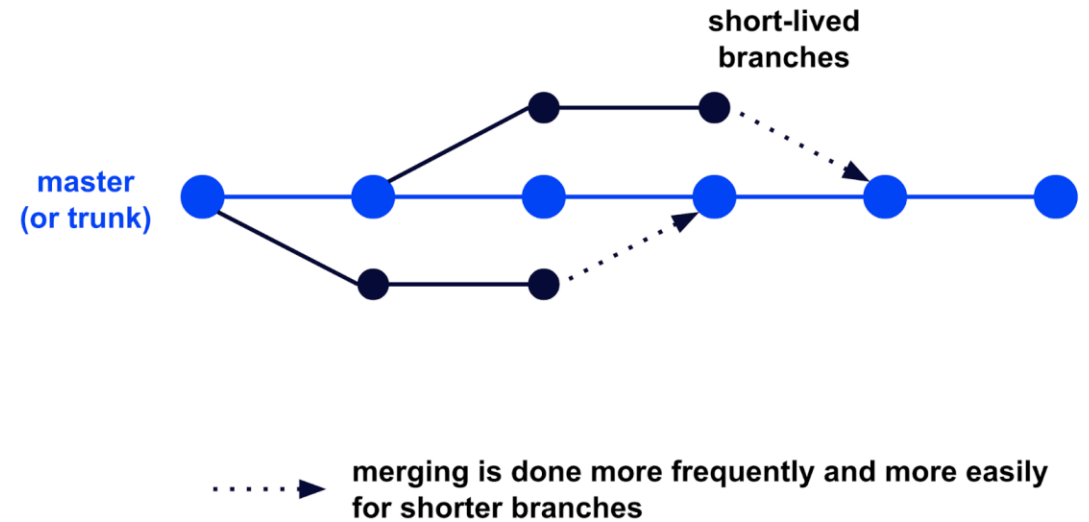
Let's try it!

- Fork the coding camp repository, and clone it locally
- Set up your local remote to track changes from the forked remote
- Create a new branch, add a file, and

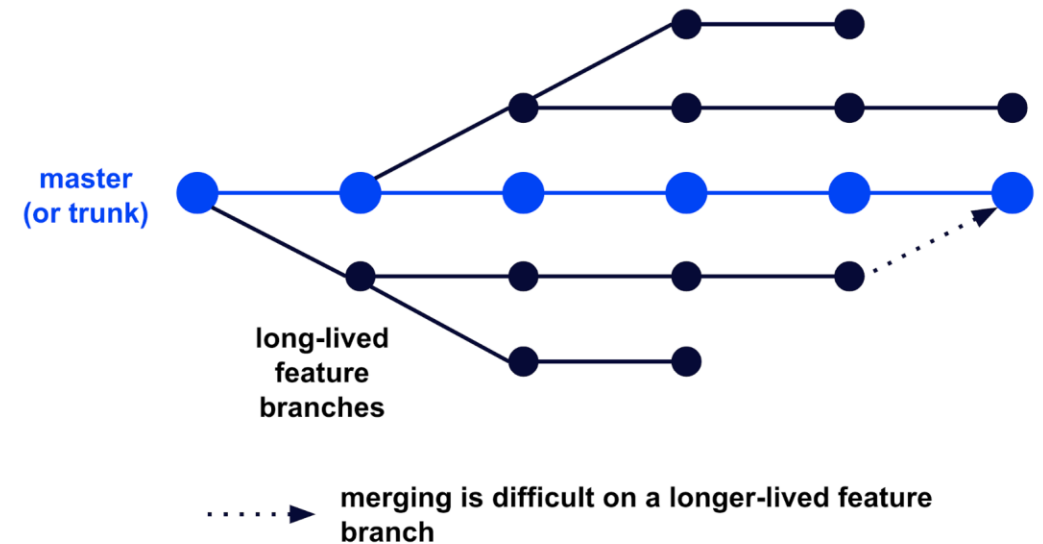
Branching

- Branches organize changes to code
- Branch for new features, bugfixes, etc.
 - Branch early, branch often!
- Branches are merged back into main
 - Main must remain stable

Trunk-based development



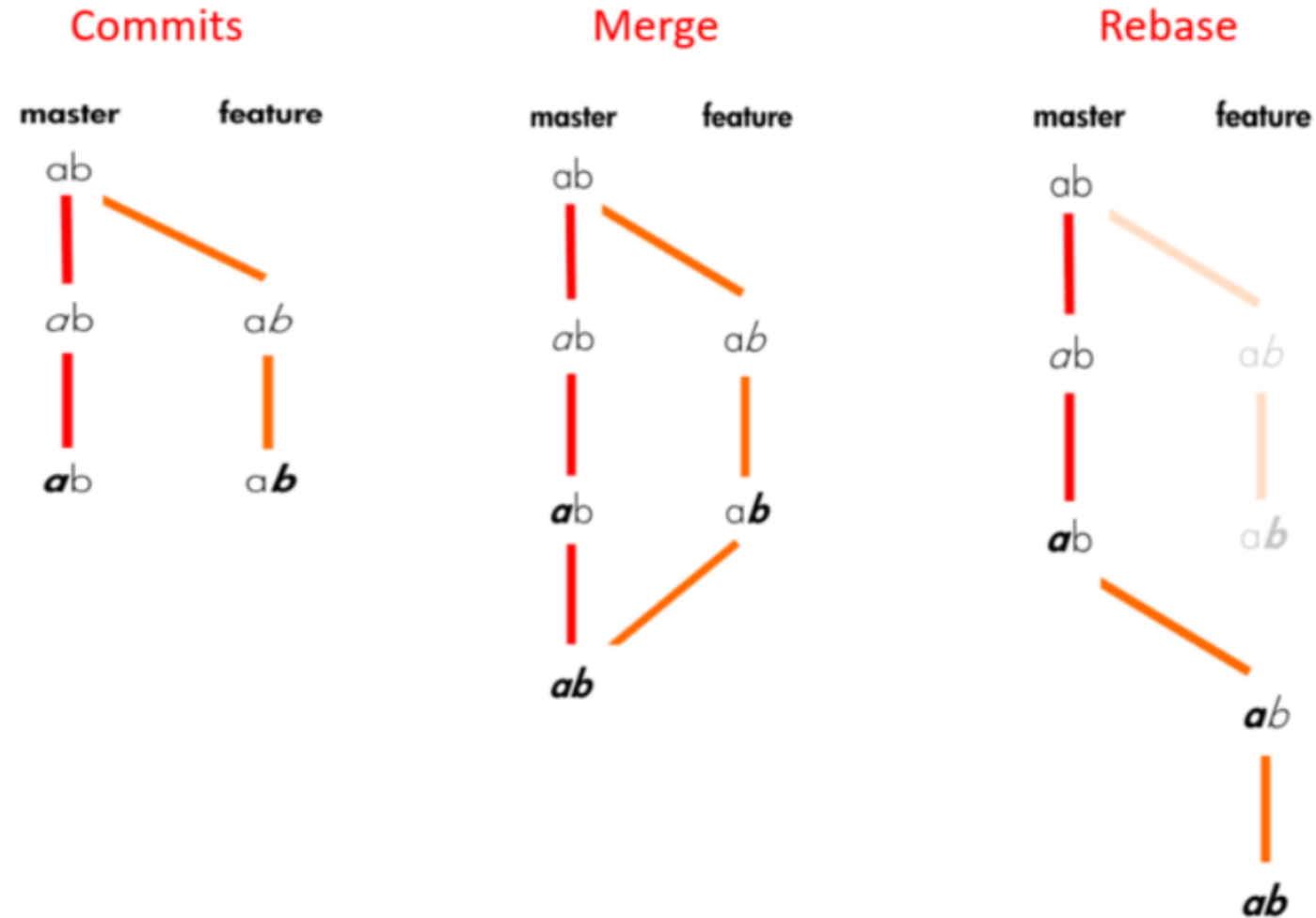
Feature-branched development



Images courtesy of Optimizely, <https://www.optimizely.com/optimization-glossary/trunk-based-development/>

Rebasing

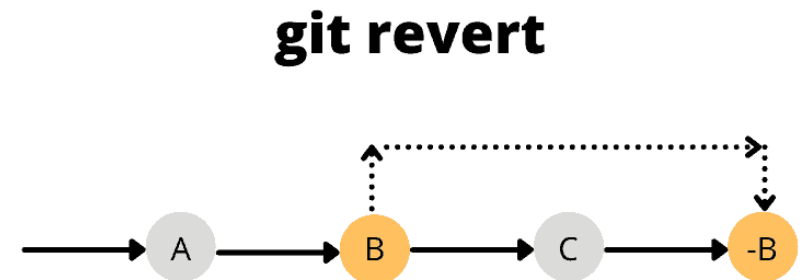
- Rebasing attaches/moves commit history
 - Looks like development in series!
- Remember, the argument of git rebase is what you want to rebase onto!
 - Ie, to move branch A onto branch B, checkout A, then git rebase B



<https://hackernoon.com/git-merge-vs-rebase-whats-the-diff-76413c117333>

Undoing things: Git Revert

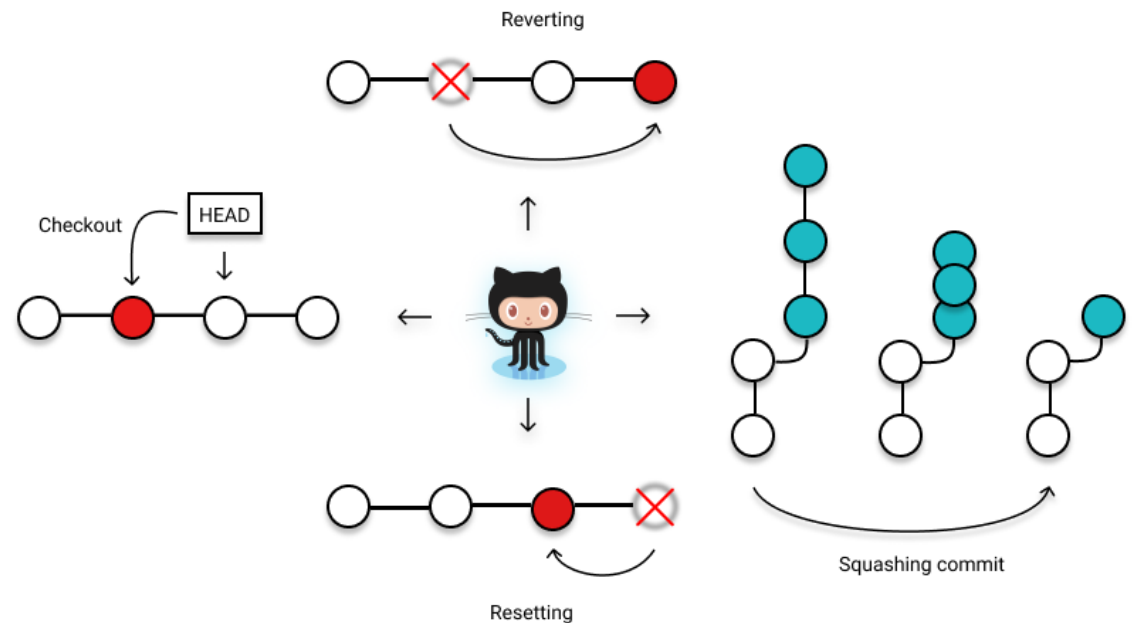
- Creates a new commit, with the contents of the old commit
- Eg: `git revert HEAD~3` => Reverts the changes specified by the fourth last commit in HEAD and creates a new commit with the reverted changes.



<https://www.becomebetterprogrammer.com/git-revert-last-commit/>

Undoing things: Git Reset

- Moves a branch backwards, as if the commits never happened in the first place
- But, doesn't work for remotes – need to use git revert
 - (You can't overwrite someone else's work)



<https://medium.com/mindorks/use-of-git-reset-git-revert-git-checkout-squash-commit-2b721ca2d2d3>

Homework

- Work through the tutorials at <https://learngitbranching.js.org/> - finish up to the "Juggling Commits" lesson