An Introduction to Git and Version Control

Presented by Yue Chen and Kenneth Steimel

How do I get git?

- Generic: Visit https://git-scm.com/downloads and download the appropriate version for your operating system
- If possible, it’s probably better to download from a package manager like homebrew on OSX, apt on debian, or choco on windows
    - However, if you don’t have one of these package managers already installed, we recommend you just install from git-scm.com
- We will be concentrating on the command line usage of git but there are good gui interfaces to git like Gitahead
- Visit github.iu.edu and create an account if you’ve never used IU’s github before.

Git what? Version control?

- Git is a type of version control system (CVS, Subversion, Mercurial)
- Version control is a system where code can be checkpointed
  - Checkpoints can be shared with other people
  - You can rewind to get back to a previous checkpoint in your code
- We won’t discuss how git differs from the other version control systems as this is ground zero
- Our tutorial draws heavily from blogs like this one Learn Git and Github: A beginners guide for complete newbies

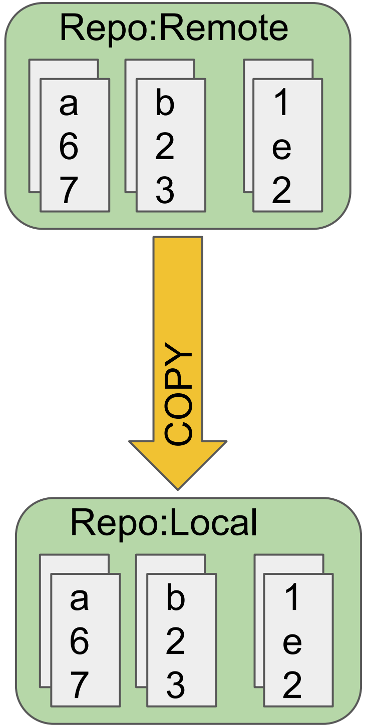
Some vocab

- Repository (repo): collection of files and folders that git is managing. Your project folder
- Github: one possible server solution for using git. IU provides an enterprise git server which has unlimited private repos
- Commit: a checkpoint in your code, it contains the state of various files and some meta information like who made the commit, when it was made etc.
  - "Like you're gathering up the files as they exist at that moment and putting them in a time capsule"
- Push: sending a commit to the central server
- Branch: The branches of your repo are like separate offshoots where the code can be developed without interfering with the main codebase.
- Merge: Combining two branches together. Can also occur if conflicting changes are made

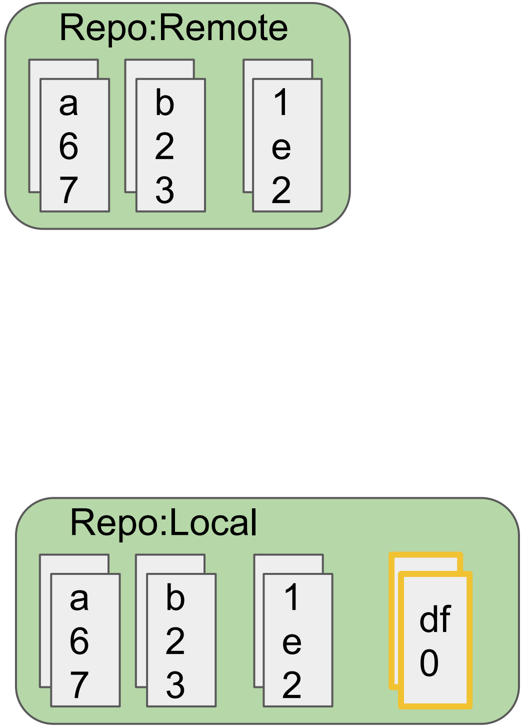
Some vocab

- Clone: Download the entire copy of the repository from the remote server
- Checkout: Go to a different commit or branch
- Fork: Copy a repo on the server (puts the new copy under your own username).
- Pull Request (PR): Submission of a request for the changes on your branch to be merged into the master branch.

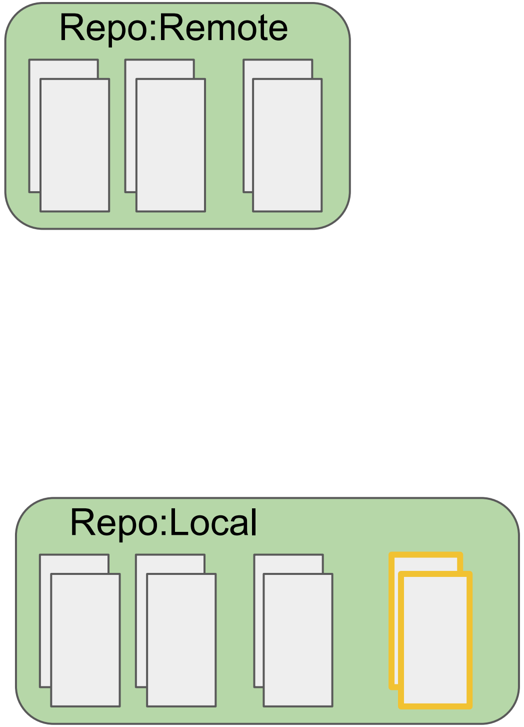
Working in git: Clone



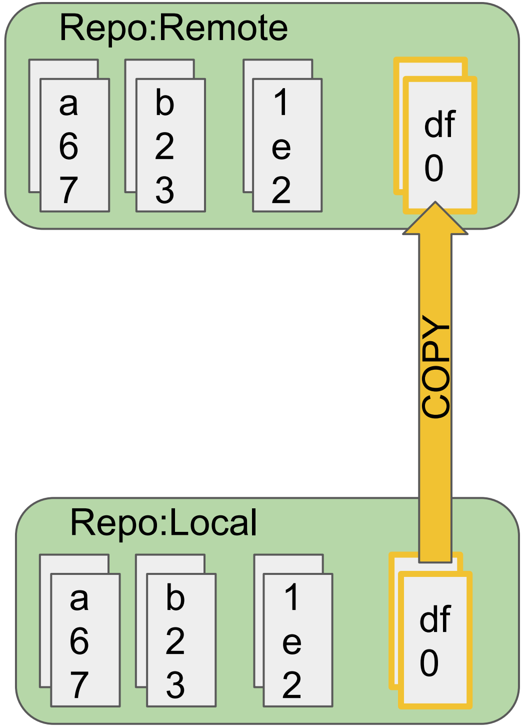
Working in git: Commit



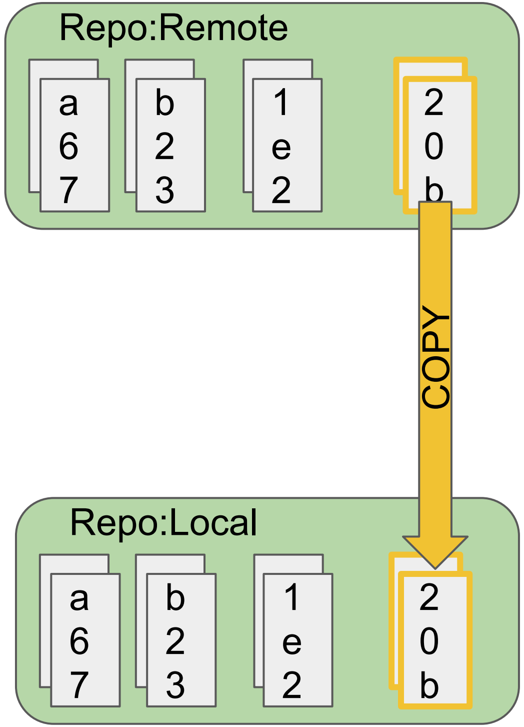
Working in git: Commit



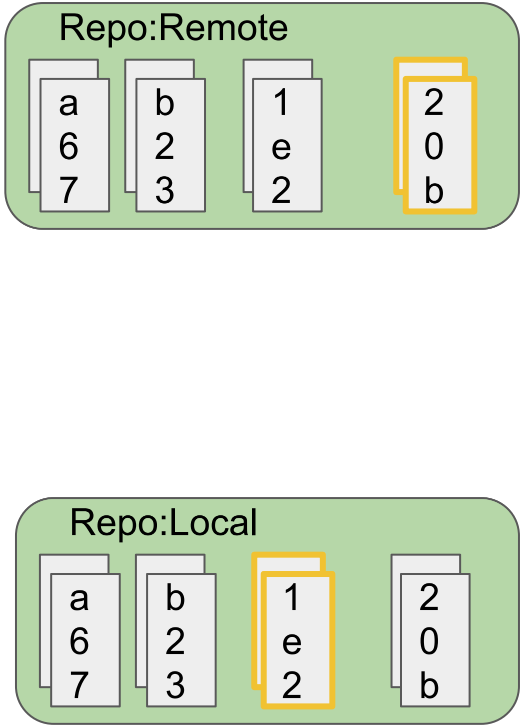
Working in git: Push



Working in git: Pull



Working in git: checkout 1e2



Practical examples

- Let’s walk through a git repo as it shows up in IU’s github

Practical examples

- Fork my repo and let’s walk through the instructions there.
- https://github.iu.edu/ksteimel/git_tutorial