

Git remotes, pull, push

- Often we need to coordinate git repositories between multiple developers
- One primary version is maintained (unfortunately currently named “master”, hopefully this is changed by git soon)
- Each developer makes their own copy of the primary repo, makes changes there, and can “push” changes back to the primary (where they can be merged into it)
- If the primary is changed, devs can pull those changes from the primary and merge them into their own local copy

Remotes continued

- The primary version may be located on a different server, in which case we identify the server and the repository we're dealing with
- Often there is some centralized organizational structure so that one person is in charge of keeping a "clean" version of the primary someplace (controlling what actually gets merged into it and when)
- The git process our department uses to distribute/collect labs and projects is one variation of this idea

Using our own copy of a repo

- If we know the url for a repo, we can use clone to create our own local copy (`git clone url`)
- We can specify that we want to be able to pull changes from some central version of the repo (by url), giving a name we'll use to refer to it (`git remote add name url`)
- We can pull from the remote (`git pull`) or push to the repo we cloned from (`git push`)
- We can specify specific branches (`git pull remotename branchname`), (`git push origin branchname`)

Comparing local to remote

- git status will tell us if we've got local commits we haven't pushed (ahead of master by N commits)
- We can fetch, but not merge in, changes from the primary repo (git fetch *reponame*)
- We can then see what's different (git diff master *reponame/master*)
- We can then merge if we wish to (git merge *reponame/master*)