

GIT (CONT.)

RECAPPING LAST CLASS

- Unstaging files
- Getting rid of changes since last commit
- Remotes
- Branching
- Things not to do (or do at your peril)

MERGING

- Last time we assumed no merge conflicts
- Merge conflicts: when 2 commits compete
 - People editing the same line
 - File both edited and deleted

RESOLVING MERGE CONFLICTS

- Manually edit files to decide what to keep
- Markers indicate who did what:
 - Between <<<<<<< and =====:
local - branch merging into
 - Between ===== and >>>>>>>:
remote - branch being merged

RESOLVING MERGE CONFLICTS (CONT.)

- Decide which to keep or combine them in some way
- Replace with whatever decided
- Remove markers
- Repeat for all conflicts in each conflicting file
- Add files after resolving merge
- Run `git commit`

RESOLVING MERGE CONFLICTS (CONT.)

- Q1. What if I just want to keep one version or the other, is there a better way?
- A1. `git checkout --ours filename` or
`git checkout --theirs filename`
`ours` = branch merging into
`theirs` = branch being merged in
- Q2. I screwed up the merge process, can I start over?
- A2. `git merge --abort` (may delete any changes in working directory)

MERGETOOLS

- Resolve conflicts by visualizing files being merged
- Many different options: meld
- Must configure git first:

```
git config --global merge.tool meld
```

```
git config --global
```

```
mergetool.keepBackup false
```

- After running `git merge` run `git mergetool`
- After resolving all in mergetool, add all files then commit

MINILAB 04