git — Quick Reference

Basics

Almost all git operations are local.

commit = save my changes locally.

push = send my saved changes to the server.

fetch = get files or changes from the server.

merge = combine stuff with my local files.

pull = shorthand for fetch & merge.

Get Help

$ git help [verb]

$ git [verb] --help

$ man git-[verb]

Configure GiT

$ git config — config help.

$ git config --list — list current config.

$ git config [option]
— show value for option.

$ git config [option] [value]
— set option to value.

$ git config --global [option] [value]
— set option to value globally.

Start a Project

From local

$ git init
$ git add [files]
$ git commit -m ‘message text’

From Remote

$ git clone [remote url]

Add, Remove, Move files

$ git add [file name] — begin tracking.

$ git rm [file name] — stop tracking.

$ git mv [file name] — move file.

Rebase

DO NOT rebase shared code.

Commit and Push

Commits are local operations!

$ git add [file name] — adds tracked file to staging area for inclusion in next commit.

$ git commit — basic commit command. Your editor will open and ask for a committ message.

$ git commit -m ‘message text’ — Commit with message.

$ git commit -a -m ‘message text’ — Skips the add step, stages all changed files and commits.

Push copies your changes to the remote repo (server).

$ git push [remote] — basic push commad. Moves your changes to the specified remote. Just $git push if working with only one emote.

$ git push [remote] [tag name] — if you create a tag for a commit it must be included in the push.

Typical workflow

$ git pull [remote] [branch] — make sure you have the latest changes.
$ git checkout [branch] — switch to the branch you want to work on.
make some changes
$ git commit -a - m ‘message text’ — update your repo as you work.
$ git push [remote] — share your changes with the group.

Merging

$ git merge [branch name] — merges named branch with current branch.

Typical merge procedure

$ git checkout [issue branch] — switch to a branch to solve the issue.
make some changes
$ git commit -a -m ‘things I did to fix this issue’ — update your repo.
$ git checkout [branch to merge into] — Typically this will be master.
$ git merge [issue branch] — merges your new code into the prod branch
resolve any conflicts, if code changes in master and your branch you will have a conflict and need to determine which code to keep.*
$ git add [file name] — marks the conflict resolved.
$ git commit — create a new commit, marks branch merged.
$ git branch -d [issue branch] — delete the branch if it is no longer needed.

*Use a tool or manually merge the documents. Conflicts are formatted like this

```
<<<<<<<  HEAD
       [code from current branch]
===================
       [conflicting code from merge branch]
>>>>>>> [branch name]:[file name]
```

Delete above or below the divider or write a blend of the two, and remove the markers. Save the file and stage it to mark conflicts resolved.

Branching

The default branch is master.

$ git branch [branch name]
— create new branch.

$ git checkout [branch name]
— switch to branch.

$ git checkout -b [branch name] — short hand, create and switch to new branch.

$ git branch — list all branches, current will be starred, **-v** for verbose output.

$ git branch --merged — list all branches merged with the currently checked out branch.

$ git branch --no-merge — list all branches that have not been merged with the currently checked out branch.

$ git push [remote name] [branch name]
— push local branch to the remote.

$git push [remote] [branch name local]:[branch name remote]
— push to remote with a new name.

$ git branch -d [branch name]
— delete branch from local.

$ git branch -D [branch name]
— force delete a branch from local.

$ git push [remote name] :[branch name]
— delete remote branch, note the space before the :[branch name].

$ git checkout --track [remote name]/[branch name]
— create a new tracking branch.

Remotes

$ git remote — show remotes for a repo, -v for verbose output.

$ git remote add [name] [url] — add a remote.

$ git remote set-url [name] [old url] [new url] — change the url a remote points to.

$ git remote rename [name] [new name]
— rename a remote.

$ git remote rm [name] — remove remote.