Git cheat sheet

See also fixing commits

Common commands

- add
  - `git add foo.py`
- checkout
  - `git checkout -b new-branch-name`
  - `git checkout main`
  - `git checkout old-branch-name`
- commit
  - `git commit -m "topic: Commit message title."`
  - `git commit --amend`: Modify the previous commit.
- config
  - `git config --global core.editor nano`
  - `git config --global core.symlinks true`
- diff
  - `git diff`
  - `git diff --cached`
  - `git diff HEAD~2..`
- fetch
  - `git fetch origin`
  - `git fetch upstream`
- grep
  - `git grep update_unread_counts`
- log
  - `git log`
- pull
  - `git pull --rebase`: **Use this**. Zulip uses a rebase oriented workflow.
  - `git pull` (with no options): Will either create a merge commit (which you don't want) or do the same thing as `git pull --rebase`, depending on whether you've configured Git properly
- push
  - `git push origin +branch-name`
- rebase
  - `git rebase -i HEAD~3`
  - `git rebase -i main`

    - `git rebase upstream/main`
- reflog
    - `git reflog | head -10`
- remote
    - `git remote -v`
- reset
    - `git reset HEAD~2`
- rm
    - `git rm oops.txt`
- show
    - `git show HEAD`
    - `git show HEAD~~~`
    - `git show main`
- status
    - `git status`

Detailed cheat sheet

- add
    - `git add foo.py`: add `foo.py` to the staging area
    - `git add foo.py bar.py`: add `foo.py` AND `bar.py` to the staging area
    - `git add -u`: Adds all tracked files to the staging area.
- checkout
    - `git checkout -b new-branch-name`: create branch `new-branch-name` and switch to/check out that new branch
    - `git checkout main`: switch to your `main` branch
    - `git checkout old-branch-name`: switch to an existing branch `old-branch-name`
- commit
    - `git commit -m "commit message"`: It is recommended to type a multiline commit message, however.
    - `git commit`: Opens your default text editor to write a commit message.
    - `git commit --amend`: changing the last commit message. Read more here
- config
    - `git config --global core.editor nano`: set core editor to `nano` (you can set this to `vim` or others)
    - `git config --global core.symlinks true`: allow symbolic links
- diff
    - `git diff`: display the changes you have made to all files
    - `git diff --cached`: display the changes you have made to staged files
    - `git diff HEAD~2..`: display the 2 most recent changes you have made to files
- fetch
    - `git fetch origin`: fetch origin repository
    - `git fetch upstream`: fetch upstream repository

- grep
  - `git grep update_unread_counts static/js`: Search our JS for references to update_unread_counts.
- log
  - `git log`: show commit logs
  - `git log --oneline | head`: To quickly see the latest ten commits on a branch.
- pull
  - `git pull --rebase`: rebase your changes on top of `main`.
  - `git pull` (with no options): Will either create a merge commit (which you don't want) or do the same thing as `git pull --rebase`, depending on whether you've configured Git properly
- push
  - `git push origin branch-name`: push you commits to the origin repository *only if* there are no conflicts. Use this when collaborating with others to prevent overwriting their work.
  - `git push origin +branch-name`: force push your commits to your origin repository.
- rebase
  - `git rebase -i HEAD~3`: interactive rebasing current branch with first three items on HEAD
  - `git rebase -i main`: interactive rebasing current branch with `main` branch
  - `git rebase upstream/main`: rebasing current branch with `main` branch from upstream repository
- reflog
  - `git reflog | head -10`: manage reference logs for the past 10 commits
- remote
  - `git remote -v`: display your origin and upstream repositories
- reset
  - `git reset HEAD~2`: reset two most recent commits
- rm
  - `git rm oops.txt`: remove `oops.txt`
- show
  - `git show HEAD`: display most recent commit
  - `git show HEAD~~~`: display third most recent commit
  - `git show main`: display most recent commit on `main`
- status
  - `git status`: show the working tree status, unstaged and staged files