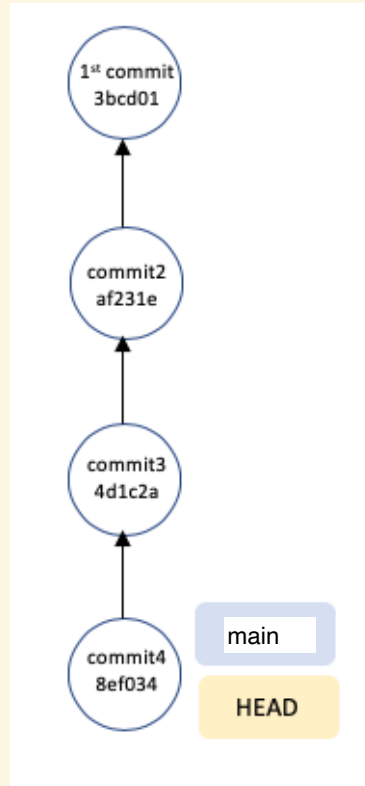# GIT BASICS - LOCAL

COMMON REFERENCES

- `HEAD` - reference to tip of current branch
- index - staging area
- SHA-1 hashes - for referencing commit hashes

BASIC OPERATIONS LOCALLY

- `git init` -> initalize new repo
- `git add` -> stage changes to file (or add untracked file)
- `git rm` -> remove
- `git mv` -> rename or move
- `git commit` -> commit staged changes

GIT COMMITS

BASIC OPERATIONS LOCALLY (CONT.)

- `git status`
- `git diff`: view changes
- `git log`: view list of commits

OOPS!

- I added (staged) a file I didn't mean to...
- Options - both unstage but keep changes in working directory
  - `git restore --staged filename` <- "new" way
  - `git reset filename` <- "old" way

OOPS! (CONT.)

- I've really screwed up a file, I just want to get rid of all my changes...
- Options:
    - `git restore filename` <- "new" way
    - `checkout` <- "old" way
        - `git checkout filename`
        - `git checkout -- filename`
        if branch exists with same name as file

- If already staged, must first unstage or
    - `git restore --staged --worktree filename`

OOPS! (CONT.)

- I've screwed up my whole working directory, can't I just go back to the last commit?
- `git reset --hard` (CAREFUL -> this deletes things)
- Or `git restore --staged --worktree .` from top level directory in repo