Version Control - Github & Git

GUI

Def: ① A system of interactive visual components of computer software
② Displays objects that convey info,
represents actions that users can take

Ex: icons, cursors, buttons

Full Name: Graphical User Interface

compare ⟷ Command Line Interface (CLI)

Git

Def: ① Distributed version-control system

② (for) tracking changes in source code during software development

Purpose → understands plain text file (machine readable)
.R/.do/.tex/.md/.txt/.csv ...
→ detect changes but can't show changes in binary files
.docx/.xlxs/.pdf/.exe/.dta... human readable

Github

- A webhosting service for all our files to be tracked
- A GUI software that provides user friendly access to git.

commands : GIT BASH

- git clone "http://..." → web address for the cloned website
download to local computer
- git add "filename".txt

- git commit -m "message"
  (haven't pushed to github.com)
- git push (push to repo github)

⇔ git pull sync the cloud updates to local computer

→ All commands need to be in the local folder cloned from Github.

① "cd" ② "dir" (Windows) = "ls" (Apple)

- git checkout → "branch name" change to branch
  * → -b "branch name" create new branches

* After creating new branches, use (git push --set-upstream origin "br")
to sync to github.com

↳ "commit id" change the local file to a formerly commited version.

(--stat): shows specific changes output

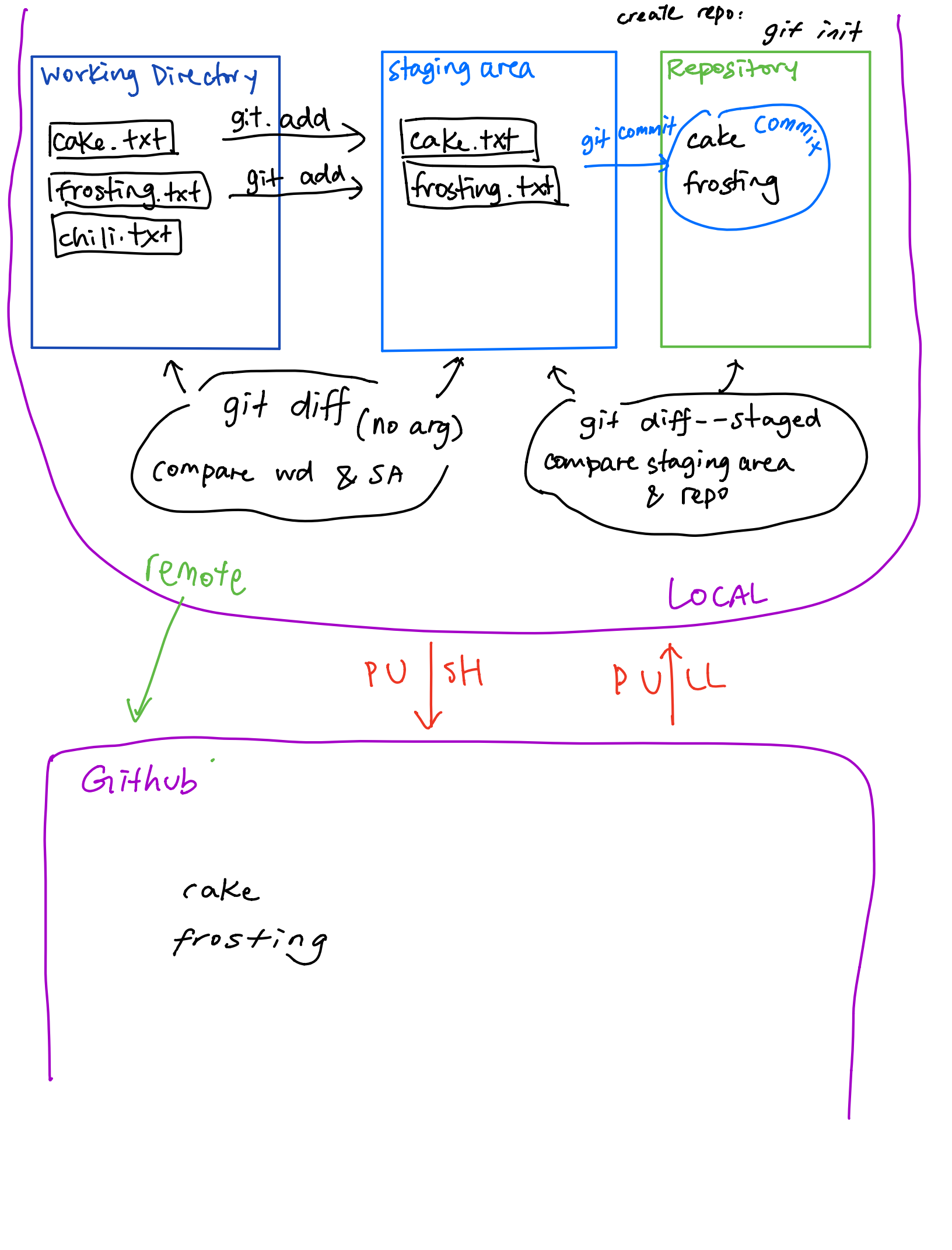
- git log >>> Shows log for each commit & the commit id
  git log --n # >>> shows last # of commits
  --graph --oneline >> tree structure
- git diff commit ID 1 ID 2
  >>> Shows difference for each version --- commit a
  green: addition red: reduction +++ commit b
- git branch → look at all the branch names
  ↳ + new name: create new branch

git checkout -b "new name"
= git branch "new name" + git checkout "new name"

- git merge master "branch name"

- ✓ needs to be in master branch.
- ✓ merge "branch" → master

- git show + commit id
  compare the difference between commit id and its parent
- git branch -d "branch name"
  (remember to merge b/4 delete so no info is lost)
  → delete the label of the branch, not necessarily the commits.

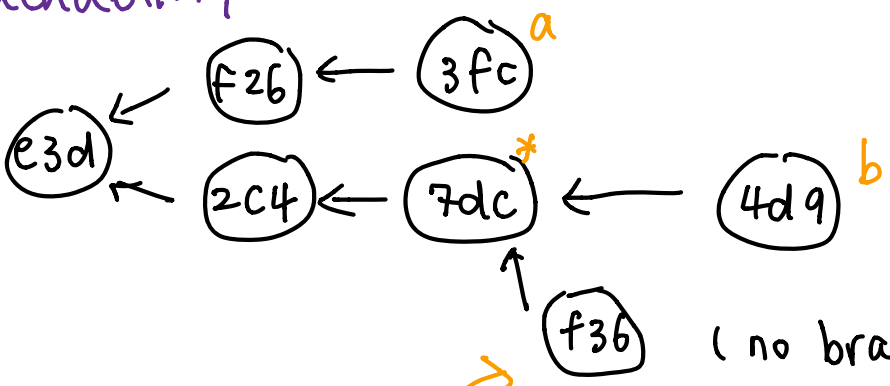

create repo:
git init
Working Directory
cake.txt
frosting.txt
chili.txt
git add
git add
staging area
cake.txt
frosting.txt
git commit
Repository
cake
frosting
commit
git diff (no arg)
compare wd & SA
git diff --staged
compare staging area & repo
remote
LOCAL
PUSH
PULL
Github
cake
frosting

Branches : switch between diff versions

"remote branch": a branch created by someone else

git remembers each commit to its parent

Reachability



(no branch here, committed thru an earlier commit.)

* Detached head error message

↳ could also create a new branch here to

Replace branch A w/ branch B (still the merge command)

- git checkout branchB
- git merge -s ours master
  --strategy = ours

"OURS" strategy:
Use everything in our current branches to replace the content in the other branch.

Making Pull Requests

- Name origin: requesting that the reviewer pull my branch to another branch to merge
- Collaboration format: one owner, multiple pull req.

How: a. checkout a new branch

b. add, commit, push (git push origin new-branch)

c. go to github.com, there will be "pull request" in the repository so that the owner can review and make comments

d. Make changes according to the comments & then repeat a-b.