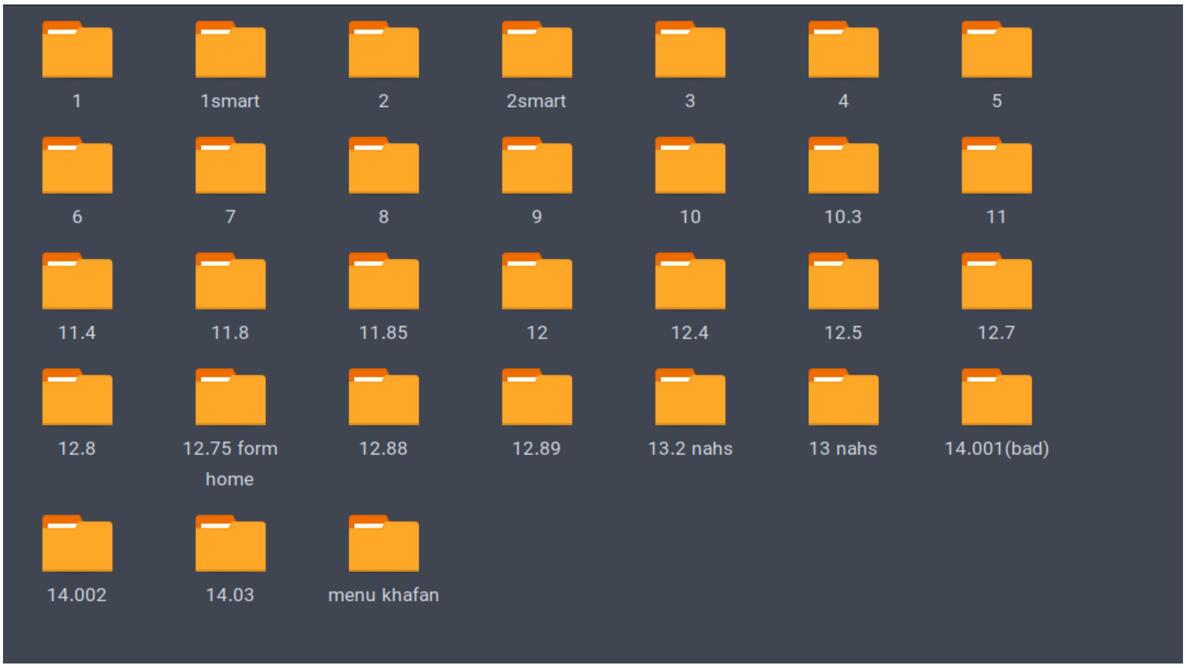
git!

> Git (/ɡɪt/)[7] is a distributed version-control system for tracking changes in source code during software development.

Table Of Contents

current problems

traditional version controlling by programmer



- copy-paste/save-as whole project after every stable build
- what if more than 1 developer work at the same time?
- which one was stable?

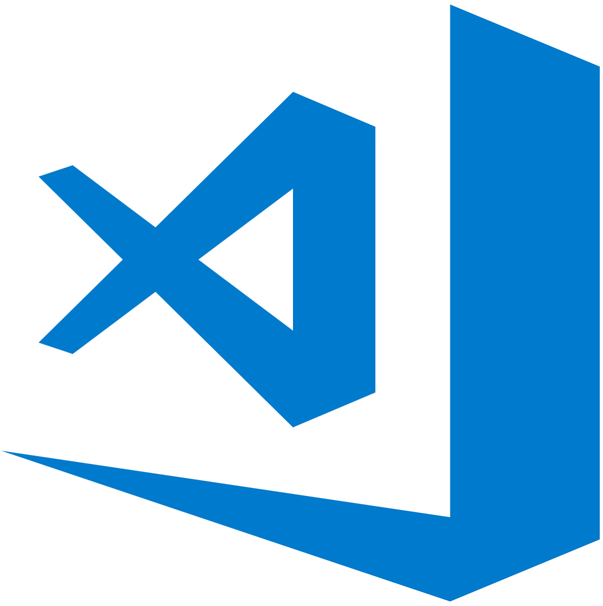
- which files are unnecessary

what is version control?

- easily management collaboration on a project
- ability to have unlimited number of developers
- easily revert back your files if something went wrong

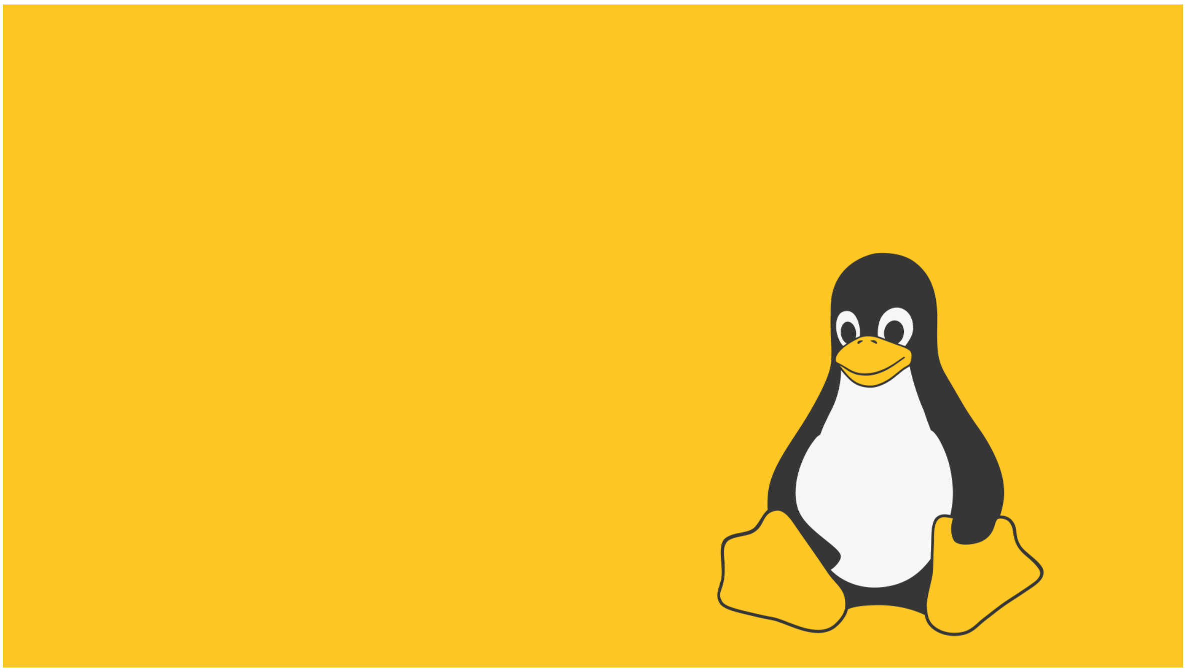
- SVN (Apache sub-version)

- Visual Studio Team Services code (by Microsoft)



- ## git

more about git

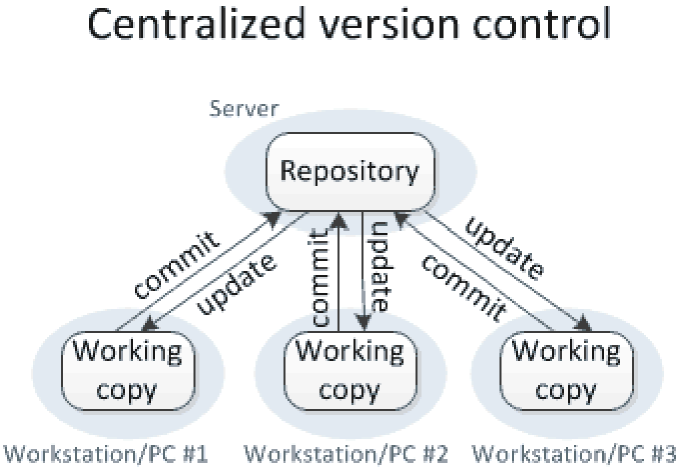


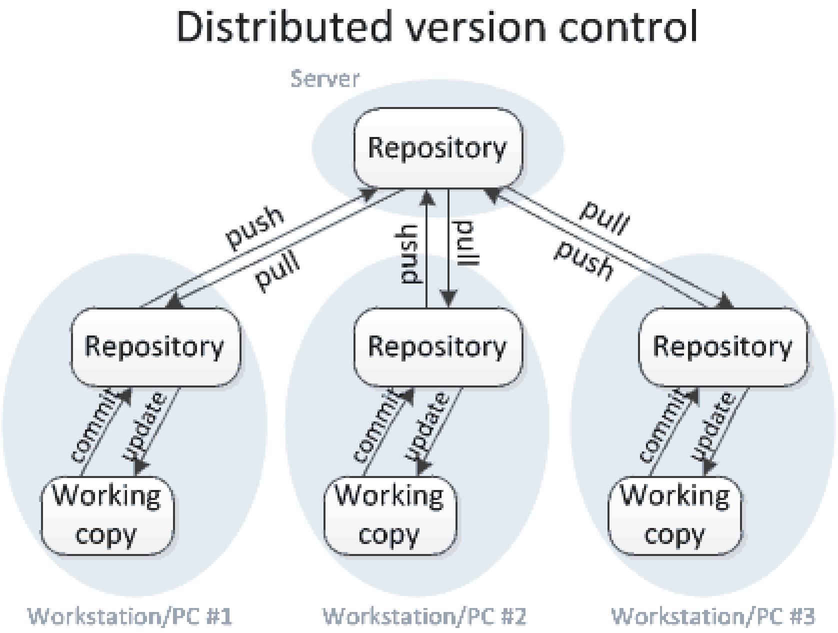
```
As of 2020, the 5.6 release of the Linux kernel had around 33 million lines of
code. (from wikipedia)
```

by Linus Torvalds

- free and open source
- distributed
- non-linear (branches)
- handle large projects efficiently

distributed vs centralized

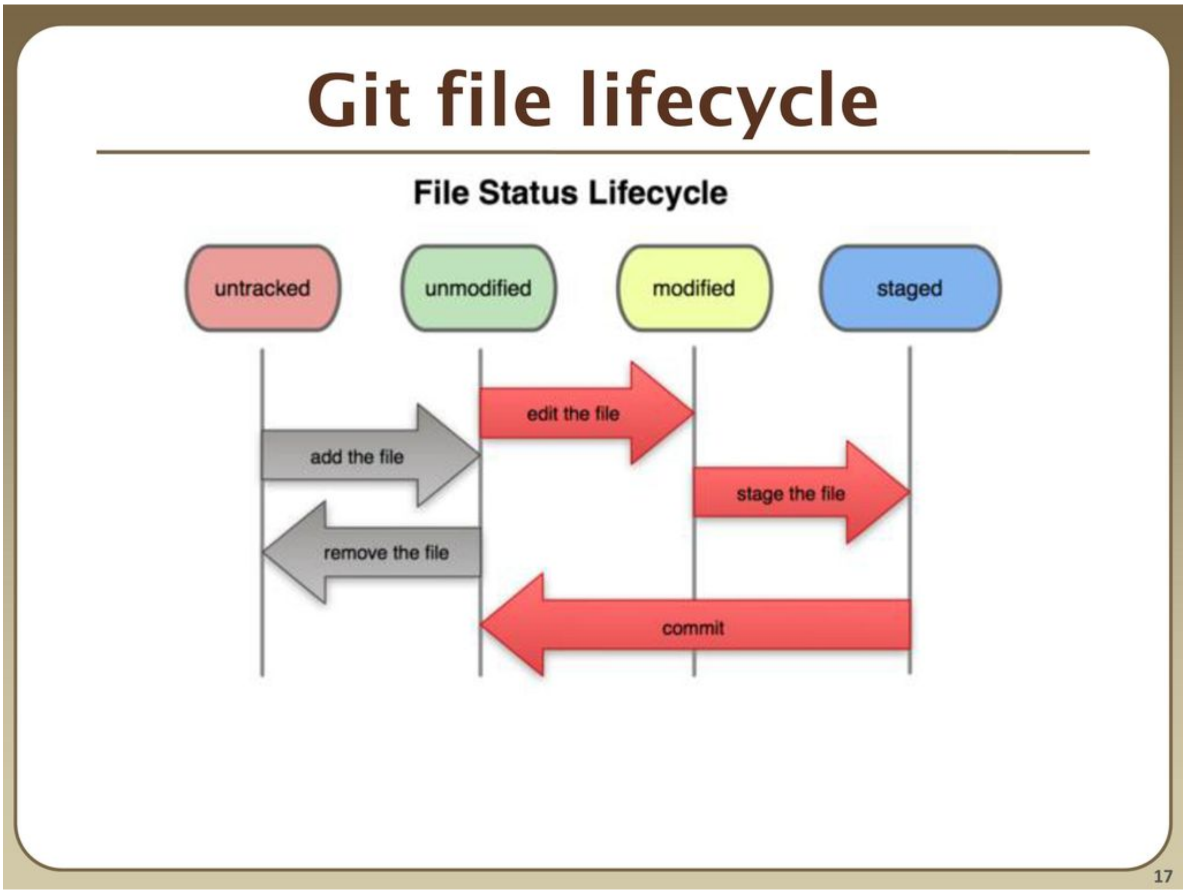




how to use git

1. search!
2. I search too
3. everybody else does search too

file status life cycle



Git file lifecycle

File Status Lifecycle

untracked

unmodified

modified

staged

add the file

remove the file

edit the file

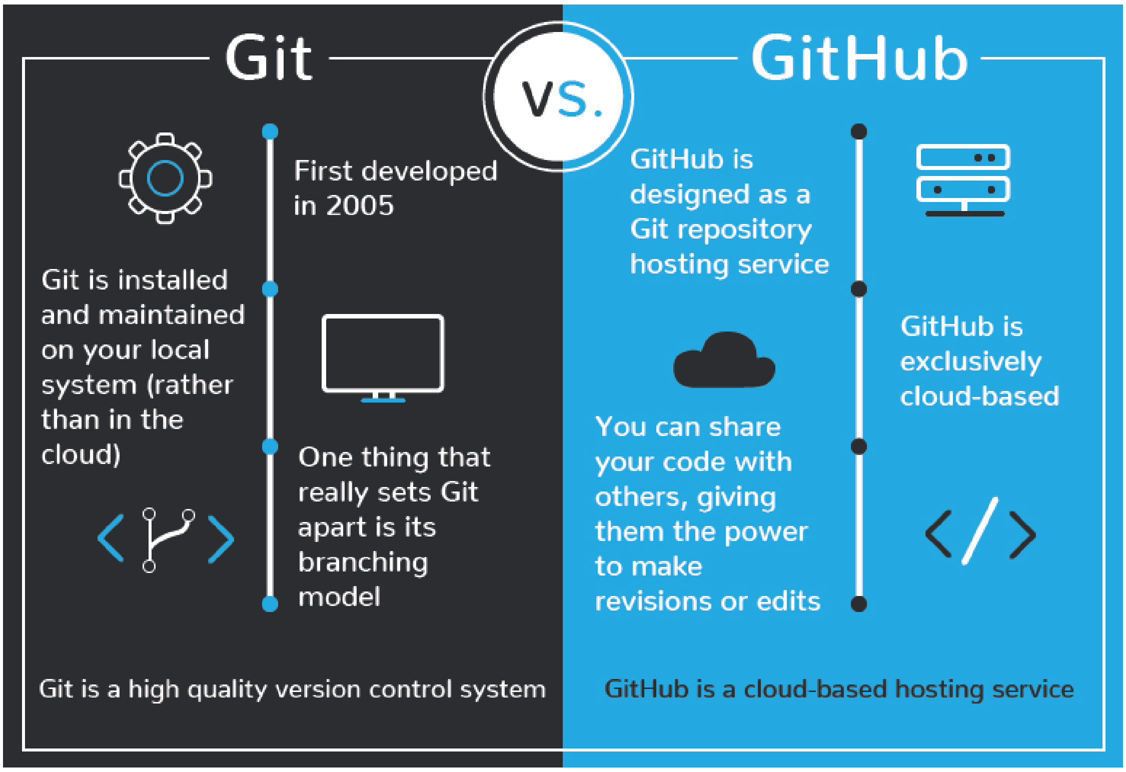
stage the file

commit

17

github

- instagram for gits
- a place to keep gits! review them, fork them, star them, issue on them..
- alternatives: gitlab, bitbucket, any other place like: CEIT gitlab



Git

VS.

GitHub

First developed in 2005

Git is installed and maintained on your local system (rather than in the cloud)

One thing that really sets Git apart is its branching model

Git is a high quality version control system

GitHub is designed as a Git repository hosting service

GitHub is exclusively cloud-based

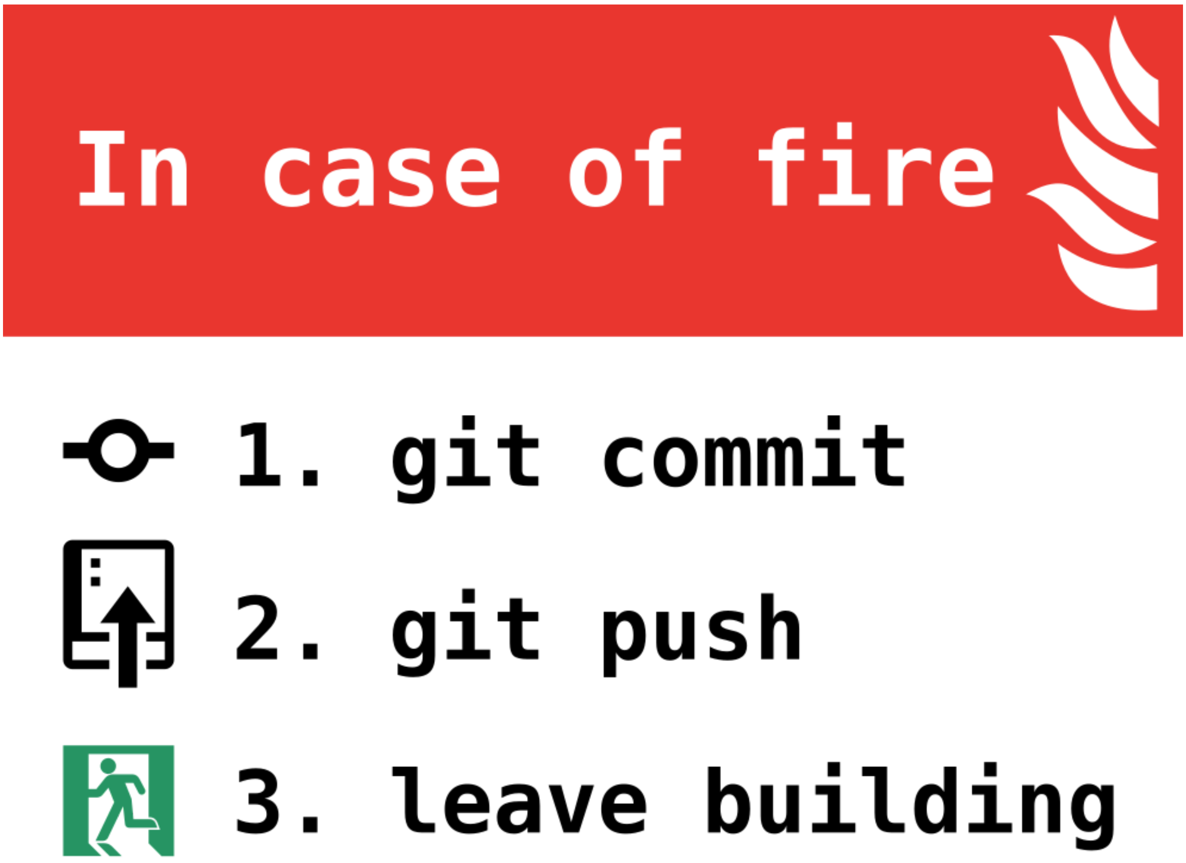
You can share your code with others, giving them the power to make revisions or edits

GitHub is a cloud-based hosting service

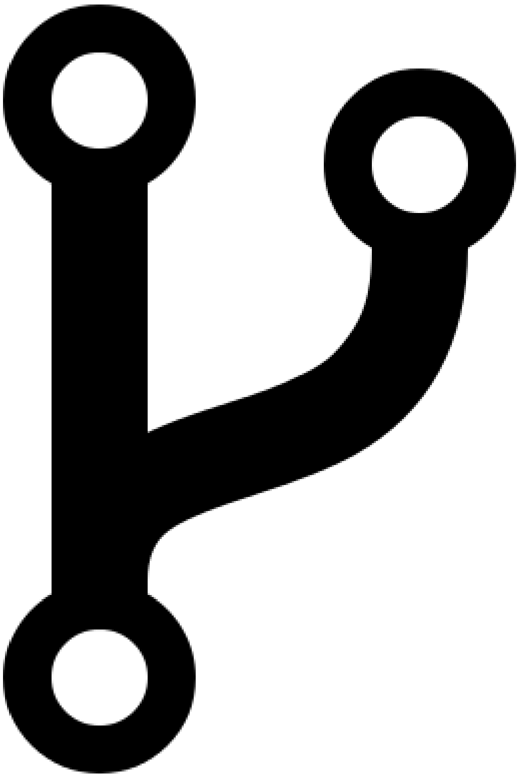
gist: some part of code to share with others

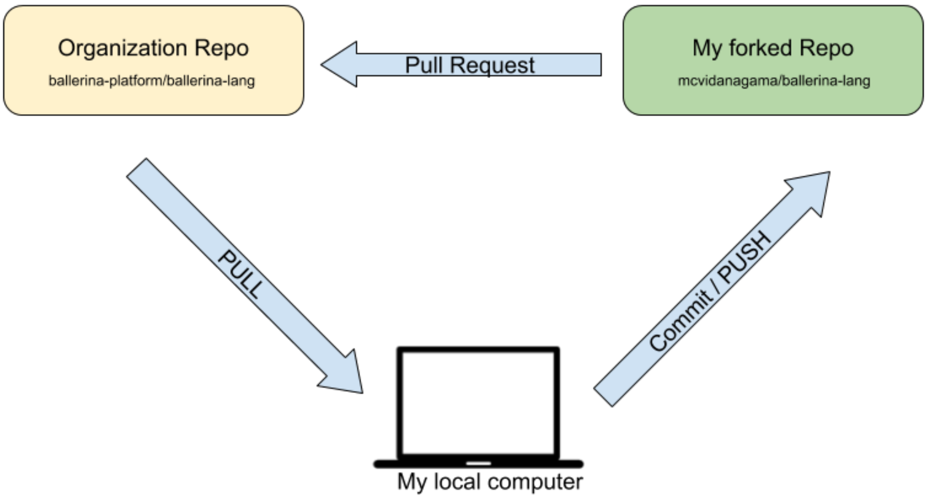
push? remote? clone?

- remote: where should i upload my gits
- push: act of uploading gits
- clone: download whole git
- pull: check for updates in the remote git

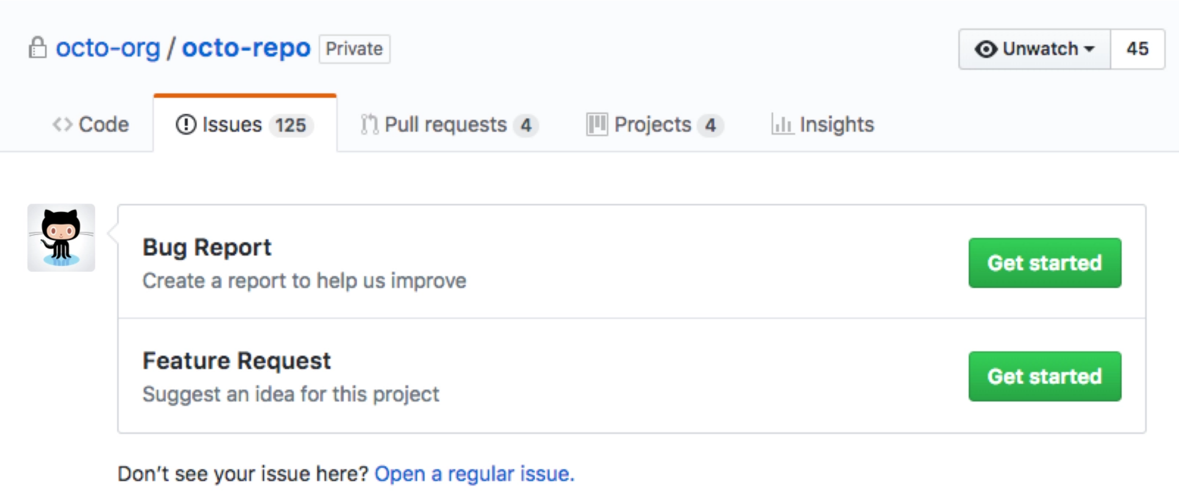


fork, PR, issue





issue, issue template



octo-org / octo-repo Private

Bug Report
Create a report to help us improve

Feature Request
Suggest an idea for this project

Don't see your issue here? Open a regular issue.

.gitignore, .git

- git: local and hidden folder that contains git internal files, don't open it!
- delete .git folder in case of removing git from project
- .gitignore: ignore these sort of files

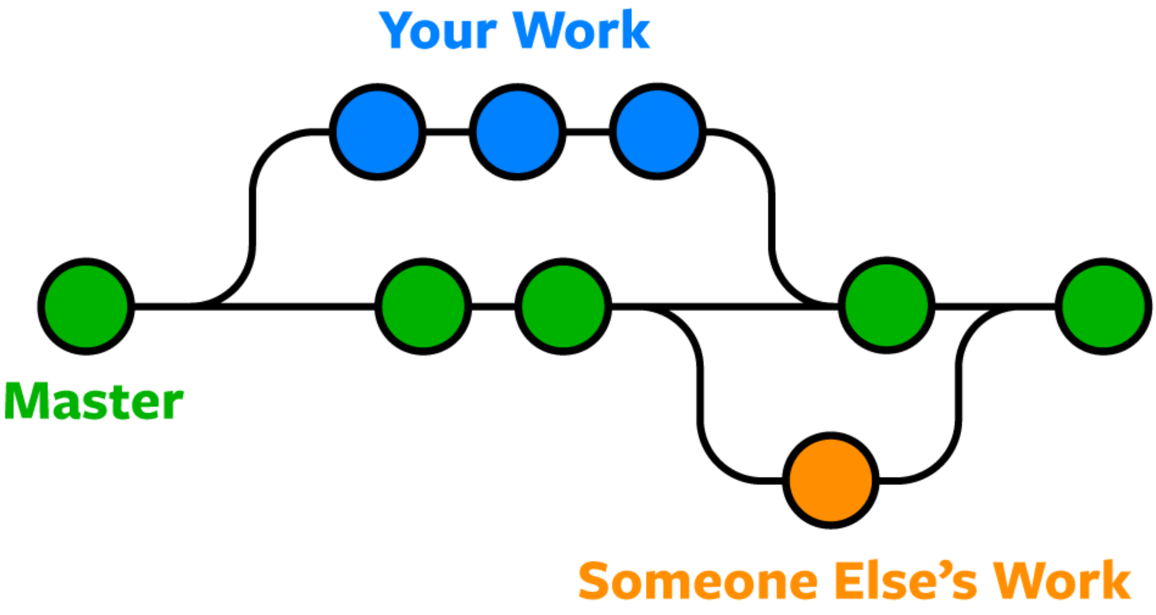
```
*.class
./.idea
__pycache__/
```

good site: gitignore.io

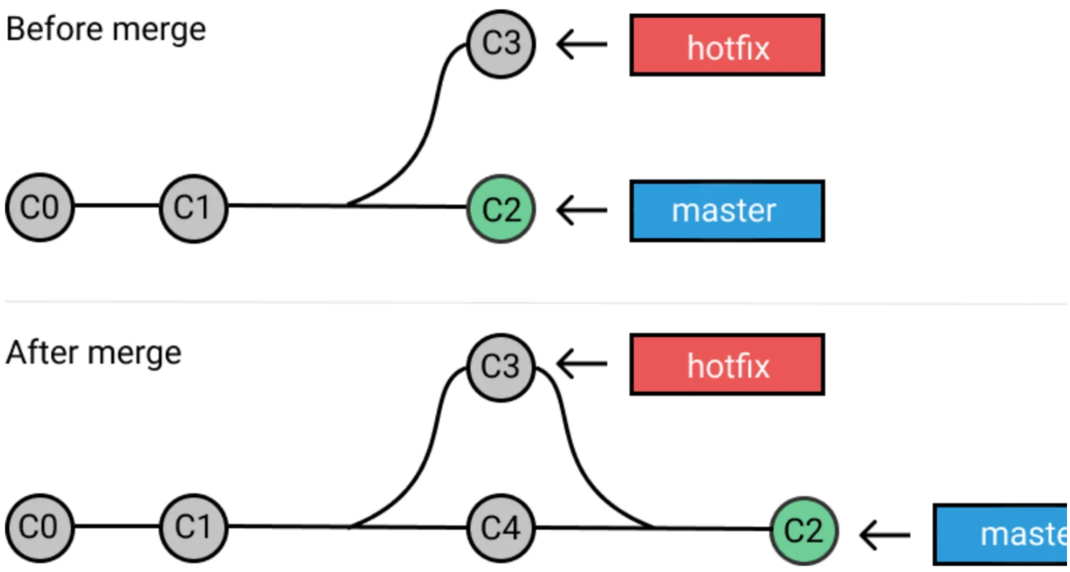
do not commit large and binary files!

why? git is source-code version control, not for keeping documents

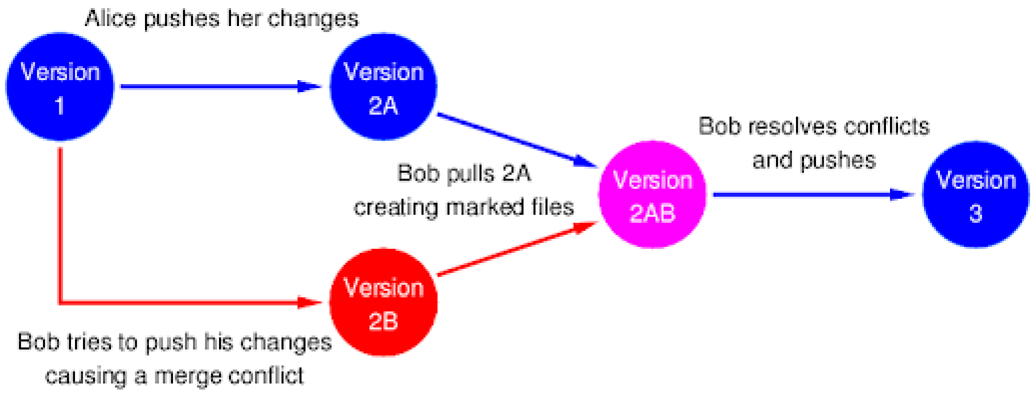
branch



and merge?



merge conflict


Alice pushes her changes
Version 1
Version 2A
Bob pulls 2A creating marked files
Version 2AB
Bob resolves conflicts and pushes
Version 3
Version 2B
Bob tries to push his changes causing a merge conflict

common commands

```
# first time initialize
git config --global user.name "Bugs Bunny"
git config --global user.email bugs@gmail.com
git init

# regulary code and commit
git status
git add -A # or git add filename
git commit -m 'commit message'

# work with remote
git remote add origin https://github.com/your-account/your-repo.git
git push origin master # from master branch to origin remote
git pull
git clone https://github.com/somebodys-account/a-good-repo.git

# see old commits and other versions
git log
git log --abbrev-commit --pretty=oneline
git checkout # change HEAD to old commits, or other branches
git diff # difference between current file and last commit (HEAD)

# eveything messed up
git reset --hard HEAD #revert to last commit
rm rf .git # get rid of git!
```

further read

- this github io page
- command by command express
- jadi's videos
- step by step
- this good slide
- tags