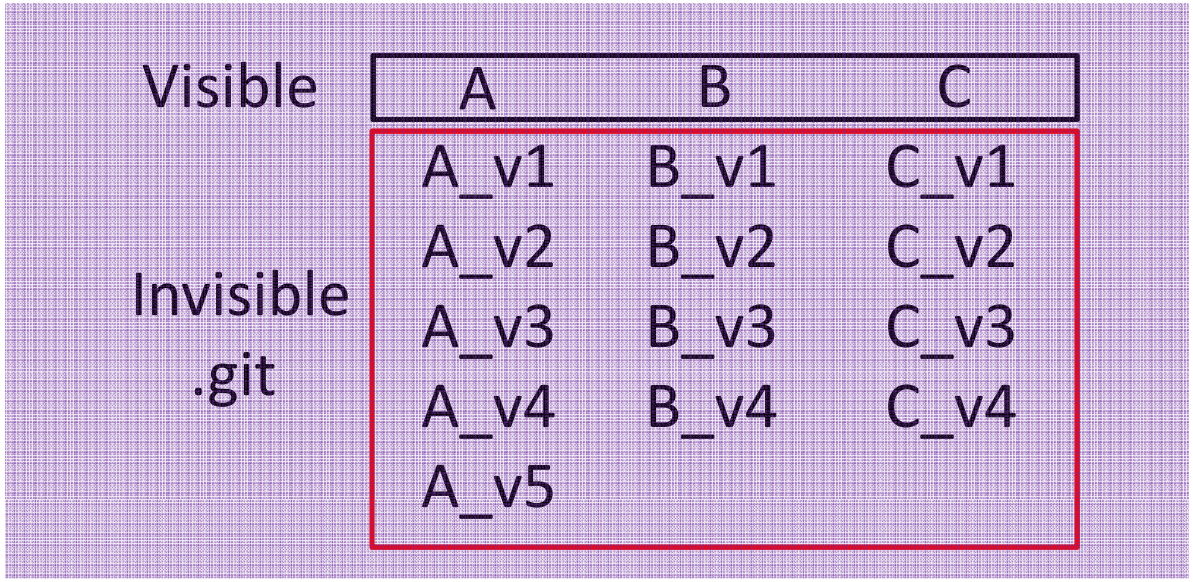
Automated Version Control

Why use git and github

- Better document of changes that you made
- Better collaboration with others
- Github can be your programming portfolio (for job applications)
- Caution to use on binary files: word
- Hard to use on pdf, and pictures(vector picture might be different)

Workflow of version control



Folder without .git

Repository with .git

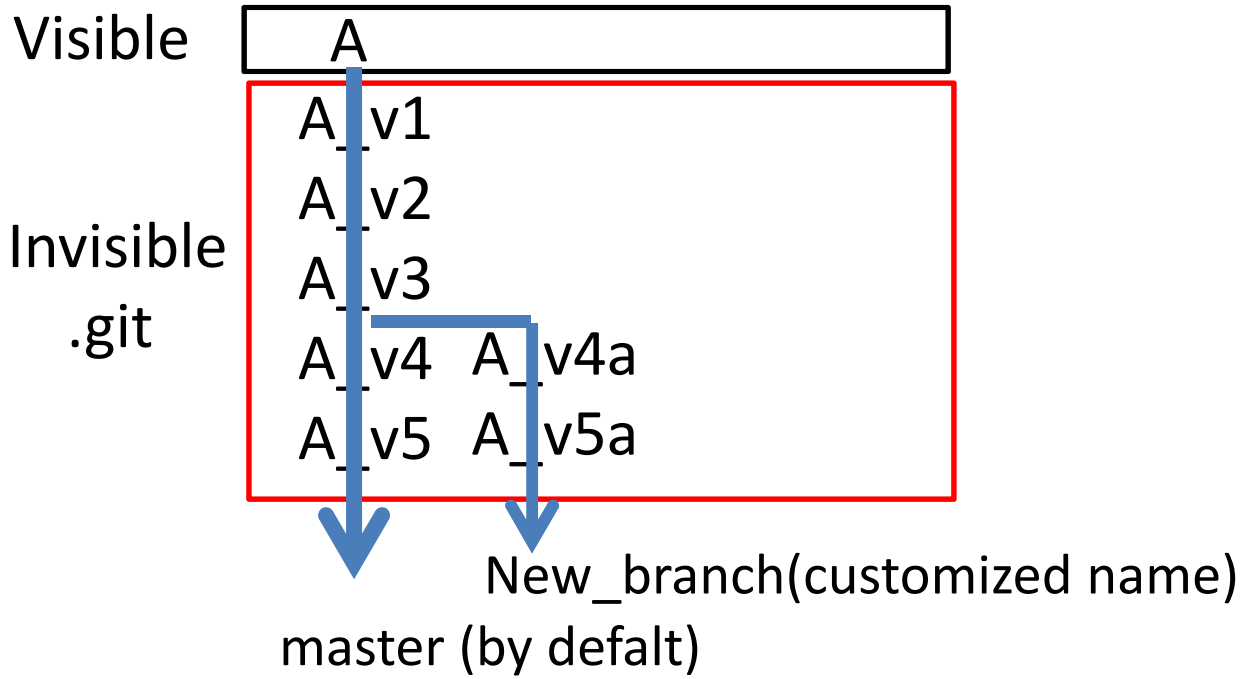
A=A_v5, B=B_v4, C=C_v4

A_v1-v5: only remembers the difference between A and A_v1-v5
But remembers who, when, why(customizable) make change

Delete A. A_v1-v5 make no sense.

Delete .git. All A_v1-v4 gone. But A=A_v5.

Workflow of version control



Branches: save space.

because computer only need to save A_v1-3 once.

syntax

git verb

to save each new version of your file

git add

git commit

git clone

git log

git status

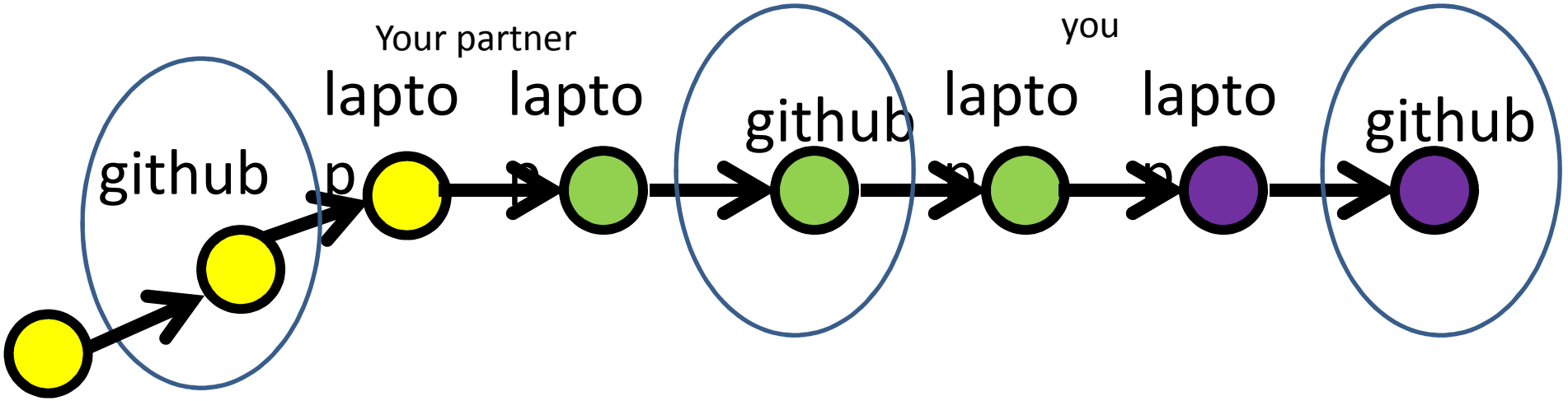
……

git add → staging

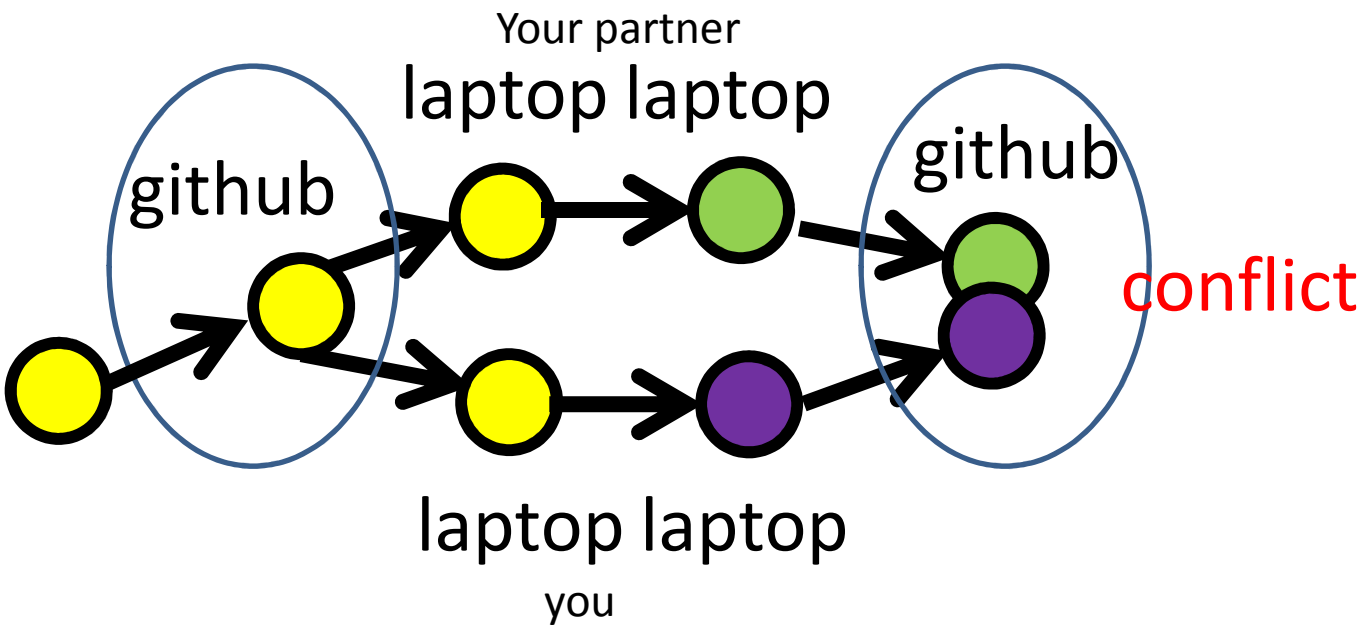
git commit → version saved

conflict

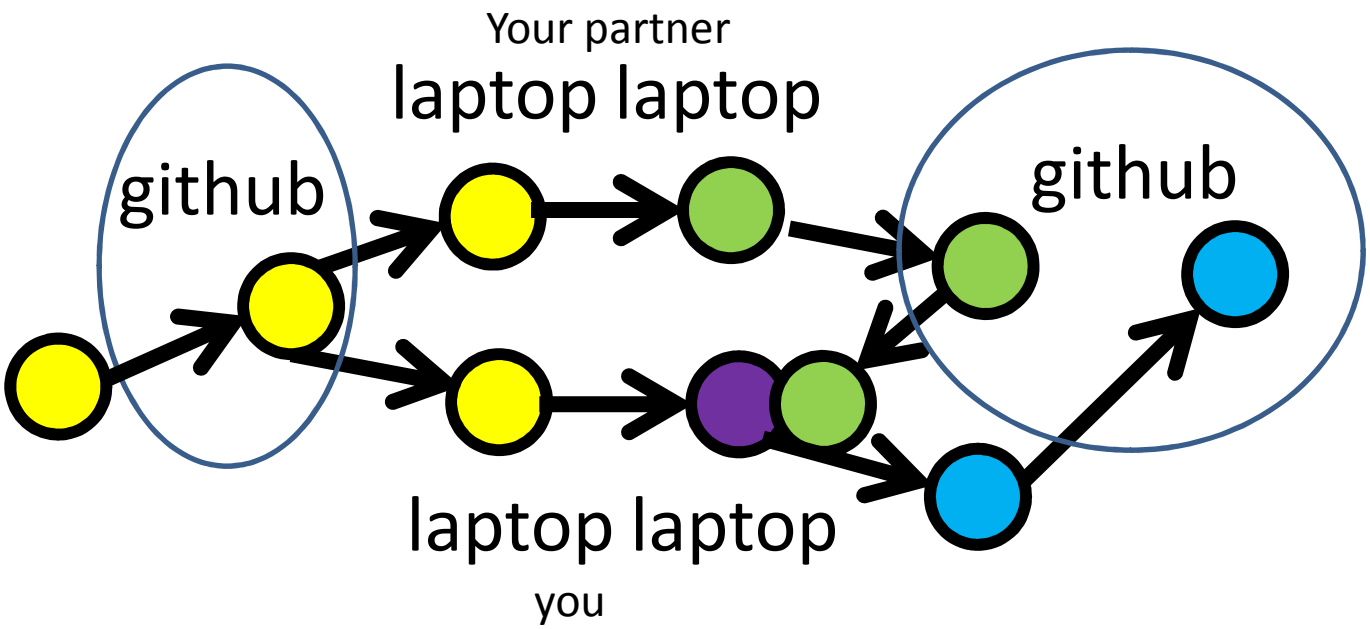
What is conflict



What is conflict



Resolve a conflict



To avoid conflict

- Remember to pull before make changes
- If conflict happens, READ HINT

```
$ git push origin master
```

```
To https://github.com/vlad/planets.git
 ! [rejected]        master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/vlad/planets.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Merge the remote changes (e.g. 'git pull')
hint: before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```