**LOCAL**

**REMOTE**

Staging area

Local repo

Remote repo

Working directory

Git push

Git add

Git commit

Git fetch(Kéo 1 nhánh về test)

Git checkout

Git pull(tải các thay đổit trên master)

Git merge

git init :Khởi tạo git trong thư mục đang đứng

git status : Kiểm tra các tình trạng các file

git add . : Thêm tất cả các file vào Staging area

git add <filename> : Thêm file đó vào Staging area

git commit -m “Mô tả commit làm việc gì”

git log : Show ra tất các commit đã commit trước đó

git show codecommit : show chi tiết về commit đó (codecommit lấy ở git log)

git diff : xem các file có midifiled, sự khác nhau giữa trước và sau

git checkout -- <filename>: loại bỏ những thay đổi của file đó trong working directory

git reset HEAD <filename> : đưa file từ staging area xuống working directory

git branch: Kiểm tra đang ở nhánh nào

gitk : Mở giao diện GUI cho git

git checkout -b <branchname>: Tạo nhánh mới và chuyển qua nhánh đó làm việc với tất cả các commit của master

git checkout <branchname>: Chuyển qua nhánh đó làm việc

Các bước để merge vào 1 branch bất kì

A 🡸 B

Step 1: git checkout A (chuyển sang A làm việc)

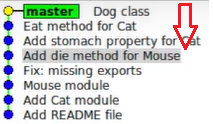
Step 2: git merge B (hợp nhất B vào A)

git reset --soft codecommit: sẽ đưa về commit chỉ định(codecommit) và trạng thái là ở staging area ,giữ nguyên các thay đổi

git reset --mixed codecommit: sẽ đưa về commit chỉ định(codecommit) và trạng thái là ở working directory ,giữ nguyên các thay đổi

git reset --hard codecommit: sẽ đưa về commit chỉ định(codecommit) và xóa các commit đang đứng đến commit chỉ định ,quay về commit chỉ định. Dùng cẩn thẩn

git revert codecommit : tạo ra 1 commit mới với loại bỏ thay đổi tại commit chỉ định , không ảnh đến các commit đang đứng đến commit chỉ định (Hạn chế dùng đến mức có thể vì các commit sửa cùng 1 file khi revert thì ảnh hưởng đến các file khác, thay vào đó tạo commit mới khác ở trên sửa lại cái nào cần sửa) Tài liệu: <https://youtu.be/8qIu9riL8w8>



Git push --force origin <branch> : sẽ ghi đè tất cả file trên nhánh <branch> tại Local Repo lên git ONLINE của nhánh <branch> đó(local có gì thì online như vậy, cẩn thận)

git remote add origin linkgithubrepo : điểu khiển repo trên github

git remote -v : Kiểm tra remote origin

git push -u origin master : khi chưa push lần nào lên

git push : đã push lên một lần nào đó

Tài liệu: <https://youtu.be/0HCbZp2VUEY>

git config --global credential.helper “cache –timeout=18000” : không hỏi tài khoản mỗi khi push lên trong 5 tiếng, thông tin tài khoản được lưu vô ram(An toàn) , còn không hỏi thì thôi.

git clone reponame : để clone các commit trên master về

git pull : để tải các thay đổi trên một nhánh bất kì

git push origin <branch> : push lên nhánh <branch>

Tạo pull request trên Github 🡺 review code 🡺 merge vào master

git brach -D <branch>: xóa nhánh khi đã hoàn thành task

git fetch origin <branchonGithub> : Kéo nhánh extend từ master về, test trên local vì review online không test code chạy như thế nào 🡺 ok rồi review changes 🡺 approve 🡺merge vào master.

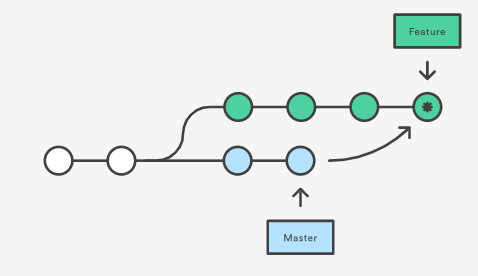
Resolve conflicts (xứ lý xung đột)

When will conflicts happen?

1. Changing the same file + same line
2. A deleted file X, B modified file X (ngược lại)

Resolve way :

1. Merge updated master to feature branch (3-way merge )
2. Resolve conflicts
3. Commit and push



git checkout feature (chuyển sảng nhánh feature)

git merge master (merge master vào feature, như trên ảnh)

References: <https://www.youtube.com/playlist?list=PLkY6Xj8Sg8-viFVtaVps_h_Emi2wQyE7q>

<https://youtu.be/K4kzvzZ7v08> Resolve conflics

<https://www.atlassian.com/git/tutorials/merging-vs-rebasing> Resolve ways