# GIT - REPOSITORY Ảnh có chứa máy tính Mô tả được tạo tự động

Remote Repository : sẽ là 1 trang chứa code nào đấy

Chúng ta sẽ CLONE về máy làm việc, khi làm việc xong sẽ commit vào máy chính và push lên để đồng bộ với Repository trên mạng. Khi có người sửa Repository này thì chúng ta dùng PULL hoặc FETCH để đồng bộ từ Repo đến cái Local

Repo: Coi là 1 dự án chứa toàn bộ source code. Repo có 2 loại là local và remote

* Local: repo trên máy tính
* Remote là trên mấy chủ, dùng một số hàm để thao tác

## Các lệnh cơ bản

* Git init: tạo ra 1 repository trong máy bạn
* Git add và git add . :
* Git remote and origin: sẽ gắn tag origin vào url này, nó sẽ nói cho git ở máy mình biết có cái remote tên origin và muốn push lên đó sẽ push theo url này
* Git push -u origin:

Commit: Khi thêm, xóa, sửa , thay đổi file sẽ là 1 commit. Commit này sẽ lưu vào repo để có thể truy ra là code này do ai thay đổi.

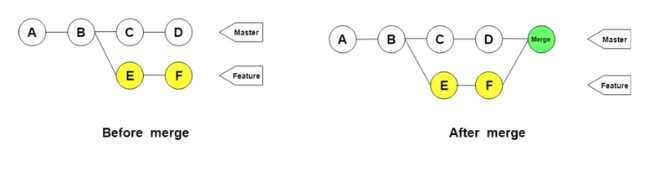
BRANCH: Khi làm source code có 1 nhánh là Master , khi muốn tách thành nhiều nhánh để code cho nó không lẫn lộn nhau thì tạo 1 nhánh mới gọi là branch. Mỗi chức năng phức tạp sẽ tạo 1 branch riêng bằng lệnh Git branch và Git checkout, khi người ta cảm thấy chức năng đó chạy ok và test ok rồi thì bắt đầu merse chức năng đó vào source code chính Master. Thường thì thay vì Merse trực tiếp sẽ tạo Pull Request (Tức là hiển thị những thay đổi của code ,chúng ta sẽ review được code , khi mà thấy code ok sẽ Accept pull request tức là merse code đó vào Branch Master) ,

* Xem các branch hiện có : git branch
* Thêm branch mới : git branch (tên branch cần tạo)
* Khi gặp lỗi trên github không có tên branch nào như này thì dùng lện: Git push –set-upstream origin (tên branch) để tạo branch trên github.
* Đổi branch: git check out (tên branch)

2. git merge (tên branch)

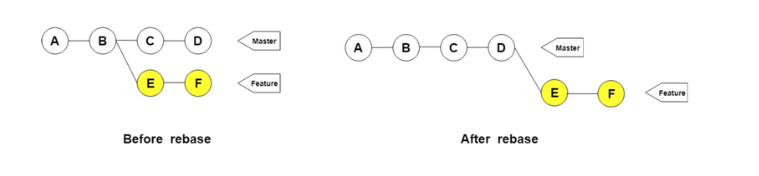
# THAY ĐỔI COMMIT

## Merge:



Khi chúng ta dùng Merge để tích hợp 2 nhánh với nhau, câu lệnh merge sẽ lấy snapshot mới nhất của mỗi branch rồi combine với nhau để tạo ra một merge commit và bỏ vào Master branch

## Git rebase:



Khi dùng rebase thì ta sẽ có 1 history dễ nhìn hơn, thằng hàng hơn. Khi dùng rebase thì lúc cần tích hợp vào nhánh master, nó sẽ đem tất cả các changes từ nhánh feature đặt lên đầu của master. Có thể hiểu là nó lấy tất cả các commit từ lúc chúng ta tách nhánh feature từ master rồi đem từng commit đó đặt lên master theo đúng thứ tự

## Git cherry-pick: