GIT TUTORIAL

Github: là một dịch vụ lưu trữ trên web giành cho các dự án muốn quản lý version nhiều người làm trong 1 project

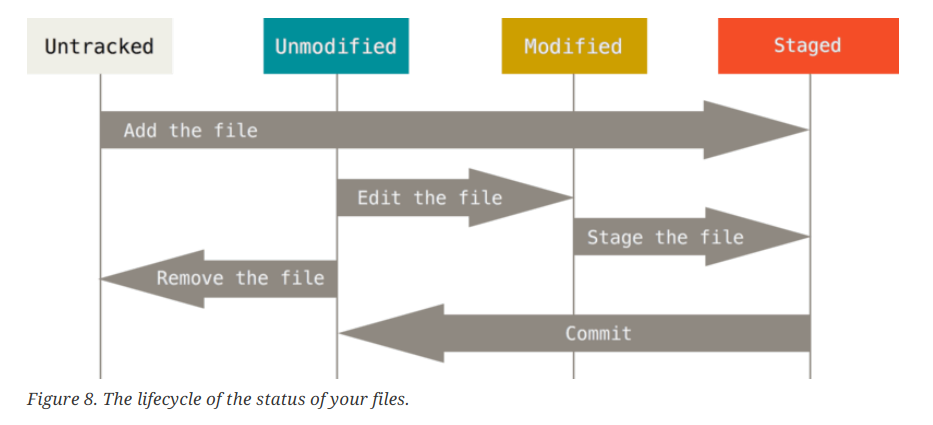
Git: là tên gọi của một hệ thống quản lý phiên bản (VSC – Version system control). Quản lý phiên bản của project lưu trên github

Repository là nơi lưu lại trạng thái của thư mục, file, sẽ chứa lịch sử thay đổi của nội dung. Tức là project đưa lên repository nó sẽ quản lý sự thay đổi của tất cả các file của project.

1. *git version* : Kiểm tra version của git
2. *Git config -- list* : Xem danh sách những gì đã config, check xem đã set username, email và password chưa.

Config username, email, password để khi mà đẩy commit lên repo thì sẽ không phải nhập lại username, email, password nữa. – global thể hiện cấu hình ở mức toàn cầu, ai cũng có thể thấy được

1. *Git config -- global user.username duong*: thiết lập username, tương tự với email và password. Khi muốn update tên thì cũng làm tương tự.
2. *Git config user.username* : Lấy ra username
3. *Git config –global –unset user.username* : Xóa bỏ username đã config
4. *Git clone url* của repository: clone repocitory từ github về local. Tạo thư mục chứa và **clone repository** đó về. Sẽ di chuyển đến thư mục đó sử dụng (**cd**) rồi thực hiện clone bằng git. Hoặc vào trong thư mục đó và mở **termial** lên
5. *Git clone url newname*. Clone repository và rename repo luôn
6. Tạo remote repository (tức là tạo repository trên github và đẩy project vào repository). Sau đó push lên remote repository
7. Git init : Khởi tạo thư mục .git sẽ theo dõi sự thay đổi tất cả các file, thư mục nằm trong thư mục đặt .git. Khởi tạp repository từ local.



* 1 file trong repository thì nó sẽ là 1 trong 2 trạng thái **tracked** hoặc **untracked**. Một file ở trạng thái **tracked** (được theo đõi 🡪ví dụ clone project về thì tất cả các file đều ở trạng thái tracked). Ta thêm file khác vào project đó chưa commit thì nó sẽ ở trạng thái **unstracked** (chưa được theo dõi).
* 1 file ở trạng thái **tracked** thì nó sẽ nằm ở 1 trong 3 trạng thái khác là: **unmodified** (không bị sửa đổi), **modified** (bị sửa đổi), **staged** (đã được đánh chỉ số)
* Khi clone project từ repository về thì các file ở trạng thái **unmodified**.

+ Nếu ta xóa file nào đi thì file đó sẽ trở về trạng thái **unstracked**.

+ Nếu ta sửa đổi file nào thì file đó sẽ ở trạng thái **modified**

**+** Khi ta **git add** file ở trạng thái **modified** thì nó sẽ chuyển sang trạng thái **staged (đánh chỉ số).** Khi file ở trạng thái **staged** ta thực hiện **commit** thì file sẽ chuyển sang trạng thái **unmodified**

* Khi file chưa được đưa lên repository thì nó sẽ ở trạng thái unstracked. Ta git add thì nó sang trạng thái **staged(đánh chỉ số**). Sau đó **commit** thì sang trạng thái **unmodified**.

1. *Git status*: kiểm tra trạng thái của file trong repository.

+Nếu ở trạng thái unmodified thì sẽ hiển thị thông báo nothing commit.

+Nếu ở trạng thái modified thì sẽ file đó hiện ra màu đỏ

+ Nếu ở trạng thái staged (được đánh chỉ số) thì file sẽ có màu xanh chờ commit.

1. *Git add .* : thêm những file mới vào, đánh chỉ số và đợi commit. Khi ta sửa đổi hoặc xóa file, thì ta đều cần phải git .add để đánh chỉ số lại chuyển sang trạng thái **staged** và chờ commit.
2. *Git add filename :* add một file vào staging area. Chuyển sang vùng đánh chỉ số chờ commit.
3. *Git commit –m “message mô tả”*: **Xác nhận chính thức rằng** **lưu lại bản sao** của các file đã được đánh chỉ số. Những file đã sửa đổi muốn commit nó lên thì phải git add . để chuyển sang đánh chỉ số.
4. *Git commit* –amend : Sẽ gộp những file thay đổi vào commit trước đó thay vì tạo một commit mới. Sau đó ta có thể sửa message. Nhấn Ctrl X sau đó nhấn Y và enter là xong và thay đổi mã commit.
5. *Git remote add name\_remote address\_remote*: Thiết lập kết nối từ local repository với remote repository.

+ name\_remote: tên của remote đặt là origin

+ address\_remote: địa chỉ của remote repository

Git remote add origin http://github.com/duonghv97/todoapp.git

1. *Git push origin <tên nhánh >* : Push commit lên một branch

VD: Git push –u origin master: push commit lên trên nhánh master. Origin là tên của remote repository.

1. Git log : Xem lịch sử của những commit thực hiện trên repository
2. Git log –oneline : Xem lịch sử của những commit ở dạng thu gọn.
3. Git log –n : Xem n commit gần nhất
4. Git reset mã commit: đưa con trỏ HEAD về commit có mã đó và những file bị thay đổi sẽ ở chưa ở trạng thái staged.
5. Git reset –soft HEAD~1 : undo về trạng thái trước khi commit

* Sử dụng khi chưa xong mà đã lỡ commit rồi. Thì sẽ undo về lúc trước khi commit. Con trỏ HEAD sẽ trỏ về commit trước đó. Các file sẽ ở trạng thái staged.

1. Git reset –hard HEAD~1 : Bỏ hẳn 1 commit từ trước và những phần chưa commit. Tức là ta sửa đổi một số file nhưng chưa commit, thì khi ta sử dụng git reset –hard thì những phần chưa commit và cả 1 commit trước đó đều mất. (mất code)
2. Git checkout