1.Source Control/ Version Control là cái vẹo gì?

* Là một hệ thống giúp thay đổi mọi thay đổi của source code
* Hỗ trợ nhiều người cùng làm việc một lúc
* Xem được ai đã thay đổi code, quản lí việc ghép code
* Hoàn tác các thay đổi, đưa code về version cũ hơn, xem xét các tính năng trước khi ghép vào source code chính, không lo mất code.

2.GIT là cái vẹo gì?

* Ra đời năm 2005
* Tác giả Linus Torvald, hỗ trợ viết Linux kernel
* Toàn bộ code và history được lưu trữ trên máy người dùng
* 3 khái niệm quan trọng: repo, commit, branch

3.Install GIT

<https://git-scm.com/download/win>

4.GITHUB là cái chi? (github.com)

Dịch vụ lưu trữ Git Repository (source code) lớn nhất vịnh Bắc Bộ

Code chung chạ với mọi anh em trên thế giới :))

Source code chính của Angular hay React,... đều ở trên Github

Có thể xem code trực tiếp, clone về máy, tạo pull reques, quản lí phân quyền

5.GIT

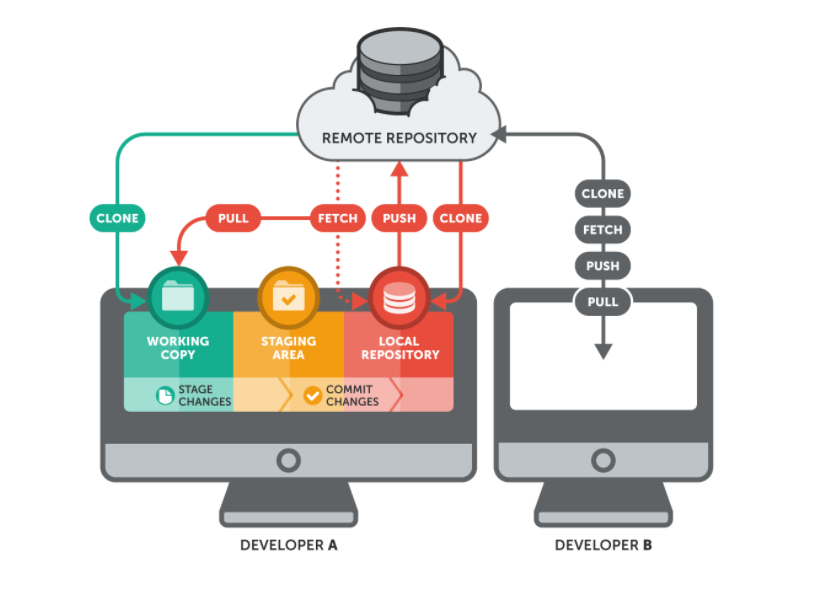
Thuật ngữ:

Working directory: là cây thư mục hiện tại mà bạn đang làm việc

Stagging area: Khi commit code thì các file đó sẽ vào stagging area (dàn dựng )

Git repository: lưu các thay đổi của các commit -> dùng git log để hiển thị

5.0.Mô hình hoạt động của GIT



A.Phần basic

5.1.git init

Tạo ra git repository trong máy

5.2.git config --global user.email ’your email’

git config --global user.name ‘your name’

Bao giờ khi cài git xong, trước khi commit hệ thống luôn yêu cầu phải config 2 dòng này. Chỉ làm 1 lần, nó sẽ tự lưu, lần sau tạo project mới ko cần config lại.

5.3.git add và git add .

Dùng khi cần khi cần thêm file, sửa file. Nó sẽ đồng bộ file hiện tại vào trong máy ở trạng thái stagging.

5.4.git commit

Mỗi lần thêm hay, sửa, xóa file,... đều phải commit file vào stagging stage, kèm theo comments

Vd: git commit -m ‘add file index.html’

5.6.git log

Xem trạng thái các commit đã tồn tại trong máy

5.7.git show + id của commit

Dùng để hiển thị nhưng thay đổi trong 1 commit

5.8.git diff

Dùng để hiển thị ra những thay đổi của file đã bị thay đổi ( modify )

5.9.gitk

Giống như git log nhưng hiển thị ở giao diện người dùng chứ ko phải giao diện dòng lệnh.

5.10.git checkout

git checkout có nhiều chức năng khác nhau

5.10.1. git checkout -- tên file

Dùng để xóa những thay đổi của những file chưa vào trạng thái stagging

5.10.2.git checkout -b <tên nhánh mới>

Tạo ra 1 nhánh mới ( có chứa tất cả file commit của nhánh master ), đồng thời chuyển sang nhánh đó làm việc

5.10.3.git checkout <tên nhánh>

Chuyển sang nhánh khác

5.11.git merge

Đưa commit từ một nhánh khác vào 1 nhánh nào đấy

Vd: master <-- A

Bước 1: git checkout master

Bước 2: git merge A

5.12.git branch

Kiểm tra xem đang có những nhánh nào

5.12.1.git branch -D <tên branch>

Xóa 1 nhánh (branch)

5.13.git reset HEAD <tên file>

Đưa 1 file ở trạng thái stagging về trạng thái chưa stagging

5.14.git reset

Đưa 1 file ở trạng thái add repository về trạng thái trước đó, và xóa tất cả commit trên đường đi của nó

5.14.1.git reset --soft <id của commit>

Dùng để thu hồi 1 commit về trạng thái stagging

5.14.2.git reset --mixed <id của commit>

Dùng để thu hồi 1 commit về trạng thái working directory

5.14.2.git reset --hard <id của commit>

Dùng để thu hồi hẳn luôn, xóa commit đó luôn. Khi dùng nên cẩn thận kẻo mất dữ liệu

5.15.git revert <tên commit>

Dùng để tạo 1 commit là đảo ngược của commit được chọn, không ảnh hưởng đến các commit trên đường đi, nhưng dễ làm thay đổi cấu trúc code -> hạn chế dùng đối với những commit cũ

2.Phần Advanced

5.1.git clone

Lấy code trên github về máy

5.2.git pull

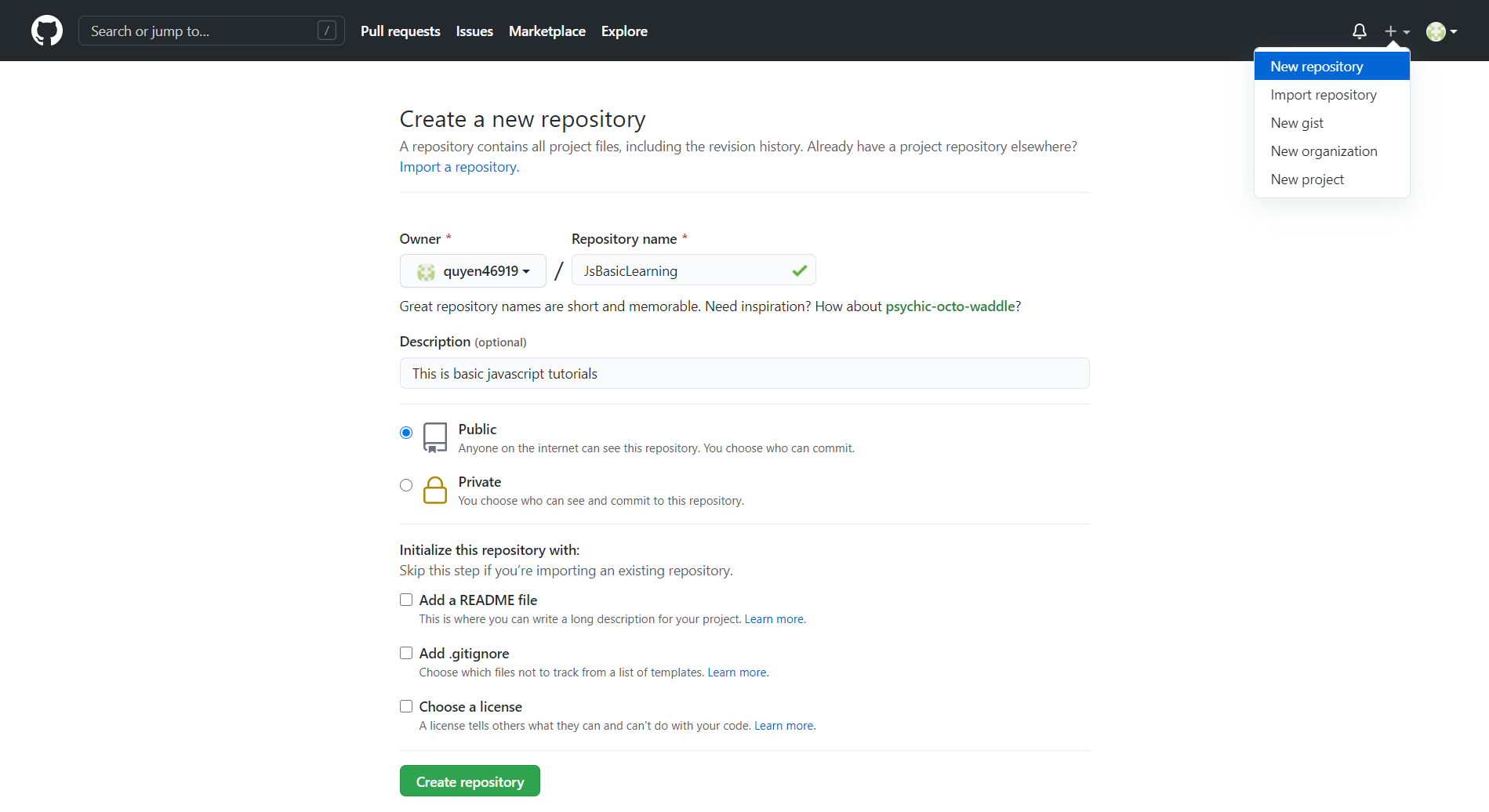
Đồng bộ code từ 1 url cụ thể trên github về máy

5.5.git push

Dùng để đồng bộ git trong máy lên trên github

6.Thực hành

6.1.Tạo git repository



6.2.Tạo git repository trong máy

