Git và github

* Git là phần mềm, cần thì download git, github là website
* Github giống google drive, để lưu trữ file cho code
* Github cho phép upload website trực tiếp và host, có cả tên miền
* Có thể sử dụng git và github để tạo website cá nhân trực tuyến với tên miền tùy chọn và miễn phí
* G&gh miễn phí, không giới hạn số lượng website, dung lượng lưu trữ, cách dùng đơn giản
* Git là hệ thống quản lý mã nguồn
* Repo trong github:kho lưu trữ
* 4 trạng thái file trên git(tại local): committed, modified, staged, untracked

+ untracked: file chưa được git quản lý, dù đã ở trong thư mục git

+ staged: sau khi add file vào git, file sẽ ở trạng thái staged; đây cũng là trạng thái chuẩn bị để commit; sau khi file chỉnh sửa(modified) file cũng sẽ vào trạng thái staged

+unmodified: file đang ở trạng thái theo dõi, nhưng file chưa có sự thay đổi

+ modified: sau khi sửa file, file sẽ chuyển sang trạng thái modified, tuy file đã được chỉnh sửa nhưng chưa commit

+committed: sau khi commit file

* Log

+git log –pretty=oneline

+git log --pretty=format:"%h - %an, %ar : %s"

* Undoing things, unstaging a staged file, unmodifying a modified file
* Các lệnh ban đầu:

+ vào thư mục, sau đó: git init để khởi tạo thư mục git

+ git status : xem trạng thái các item trong thư mục git

+ git add filename.txt, git add - -all, git add \* : thêm file vào cho git quản lý

+ git commit filename.txt -m “commit file”, git commit -m “commit files” : commit file hay files

+ git log : xem lịch sử commit

+ git commit - -amend : commit file hiện tại chưa commit, sau đó gộp thông tin các file commit này vào thông tin lần commit gần nhất: Undoing things

+ git log --stat : xem lịch sử commit, bao gồm các file đã commit, sau đó nhấn Q

+ git reset head filename.txt : unstage 1 file đã commit : unstaging a staged file

+ git checkout - - filename.txt : trả lại dữ liệu file trước khi thay đổi: unmodifying a modified file 🡪 cẩn thận vì sẽ làm mất code

* Khi đã commit rồi thì đều có thể recover lại được
* Remote Repository

+ git clone githublink : tải source về từ github

+ sau đó, cd để vào thư mục mới tải về

+ git remote: kết quả là origin: clone thành công

+ git remote -v: cho biết tên rút gọn(origin) và đường dẫn của remote repository

+ fetch: lấy dữ liệu về, push: đẩy dữ liệu lên

+ đổi tên đại diện: nếu là thư mục mới, cần git init để kết nối thư mục với git, sau đó : git remote add shortName githublink (giả sử shortName = phongT)

+ git fetch phongT : cập nhật dữ liệu mới từ remote về local, chứ không merge

+ git push testP main: đẩy dữ liệu lên server, testP là local, main là bên server, hiện tại chỉ có 1 branch

+ git pull phongT main: lấy dữ liệu từ server về, phongT là local, main là server, nếu từ ban đầu đã clone về thì dùng: git pull testP, không cần chỉ ra branch main

+ khi không thể push được: vì local là branch master, server là branch main, cần : checkout -b main : để đổi tên branch sang main, sau đó mới push

+ git remote show phongT : xem thông tin về repo server

+ git remote rename origin Ptest : đổi tên remote origin thành Ptest

+ git remote rm testP : bỏ đi remote testP

* Tagging : đánh dấu 1 commit bằng 1 tên nào dễ nhớ, để dễ tìm kiếm sau này

+ (tạm bỏ qua, coi sau)

* Tạo alias cho các lệnh(tạm bỏ qua)
* Branches

+ mỗi lần commit, git sẽ lưu lại snapshot

+ git branch: liệt kê danh sách các branch

+ git branch phong: tạo nhánh có tên phong

+ các nhánh tạo như vậy sẽ base trên commit cuối cùng của main

+ log - -oneline - -decorate: xem lịch sử các nhánh, xem HEAD trỏ vào đâu

+ git checkout phong: chuyển con trỏ head sang nhánh phong

* **Sau khi commit trong vs, cần push dữ liệu mới đưa lên server**
* Phân nhánh và merge nhánh trong git

+ git checkout main: chuyển sang nhánh main, từ nhánh main, thực hiện: git merge issue: merge nội dung từ nhánh issue sang nhánh main

+ khi gõ lệnh git push phongT, gặp lỗi :

fatal: The current branch main has no upstream branch.

To push the current branch and set the remote as upstream

Thì thực hiện lệnh

**Git push - -set-upstream phongT main** để kết nối branch main của local và main của server,

Sau đó thực hiện

Git push phongT thì nội dung mới sẽ được cập nhật lên server

+ git push - -set-upstream phongT issue: nếu trên server branch issue chưa tồn tại thì lệnh này cũng sẽ tạo branch issue trên server

* **Basic merge conflicts**

+ do 2 người ở 2 máy khác nhau cùng sửa 1 đoạn code, nên xảy ra conflict, cần developer merge code, vì git không biết merge thế nào, đoạn nào thêm, đoạn nào bỏ.

+ cần thiết lập tool merge ở file config

+ git mergetool : mở merge tool lên để merge 2 file