**KIẾN THỨC CƠ BẢN VỀ GIT**

1. **Source Control**

* Hệ thống giúp lưu trữ và quản lí source code
* Revert các thay đổi, đưa code về các version cũ hơn (không sợ mất)

1. **Git**

* Respiratory: một dự án (folder) chứa toàn bộ source code
* Commit: tập hợp các phiên bản cũ hơn (snapshot) của code
* Branch: các nhánh chức năng cụ thể

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

* Git hub: dịch vụ lưu trữ online respiratory

1. **Các lệnh git cơ bản**
2. **Sơ đồ làm việc của git**

A diagram of a computer system

Description automatically generated

**Giải thích:**

B1: Ta sử dụng các câu lệnh để đưa source code, file vào respiratory trên máy (được tạo ra từ câu lệnh git init)

B2: Đẩy lệnh từ respiratory lên git hub

B3: Ta có thể thực hiện chỉnh sửa và đồng bộ giữa local và remote.

1. **Các câu lệnh git: Ta tiến hành mở cmd (terminal) và tiến hành nhập lệnh cho git**

|  |
| --- |
| **Tạo respiratory trên máy (local)** |
| * Đầu tiên, ta mở cmd tới thư mục cần tạo respiratory   git init: tạo ra 1 respiratory của thư mục đó  git status: kiểm tra tình trạng của respiratory hiện tại  Mở rộng: git status <ten file>: kiểm tra tình trạng hiện tại của 1 file |
| **Đưa source code vào respiratory** |
| git add <ten file>: đưa file đó vào respiratory  git add . : thêm tất cả các file đó vào respiratory   * Nếu các file đó cùng nằm trong 1 thư mục thì push lên git hub cũng là 1 thư mục chứa các file đó |
| **Snapshot cho source code** |
| git commit -m “message”: thực hiện snapshot cho source code |
| **Đẩy source từ respiratory lên github** |
| git remote add <r.resiratory\_name> (origin thường là mặc định) <paste-the-copied-url-here>: kết nối với git hub từ xa  Mở rộng: ta có thể xem có phải origin không bằng lệnh git remote -v  git push -u <r.respiratory\_name> (origin) <branch namemaste> (r / main)   * Sau khi đã thực hiện thiết lập, sau này ta chỉ cần thực hiện git push (cập nhật cho respiratory)   **git push origin master** |
| **Kiểm tra tình trạng trang respiratory** |
| git status: Kiểm tra tình trạng của respiratory của thư mục hiện hành  git log: hiển thị thông tin của các commit trong thưc mục đã chụp (ta chú ý tới mã hash của commit)  git log –-oneline: hiển thị dưới dạng 1 dòng  git log –-graph: hiển thị dưới dạng đồ thị |
| **Đồng bộ hóa git xuống máy** |
| git pull <r.respiratory\_name> <branch name>: đồng bộ hóa xuống source code   * Lưu ý: Ta chỉ có thể đồng bộ khi code ở remote thay đổi và code ở local không đổi và với mỗi commit, chỉ có thể pull 1 lần.   Nếu code ở local và remote cùng thay đổi thì sẽ xảy ra conflict  **Cách giải quyết:**  B1; git pull origin master: đưa code về máy  B2: git add . git commit -m “mess” git push origin master  B3: git pull origin master: Chọn phiên bản chấp nhận được Vscode đề xuất  B4: git add . git commit -m “mess” git push origin master: để cập nhật lại code thoát conflict rồi ta làm việc tiếp |
| **Tải tệp git xuống máy** |
| git clone <paste-the-copied-url-here> |
| **Chia nhánh** |
| Đầu tiên, ta phải tạo nhánh  git branch [ten\_nhanh]  git branch: xem các nhánh và nhánh hiện tại đang ở  Chuyển nhánh  git checkout [ten\_nhanh]  Các lệnh thao tác tương tự nhưng git origin [ten\_nhanh]  Gộp nhánh   * Đưa về nhánh master: git checkout master * git merge [ten\_nhanh] |
| **Quay lại trạng thái trước đó** |
| git checkout <commit – hash – code>   * Ta có thể tìm hash – code trong lệnh git log |

1. **Chức năng git Pages**

* Dùng để host một trang web
* Mặc định sẽ chạy file index.html và style.css trong project

**Thực hiện**

B1: Upload project lên git Hub. Đặt tên file là index.html và style.css

B2: Setting -> Pages -> Branch -> Root / master

B3: Quay về phần Code. Chọn vào dấu chấm vàng để xem git hub bắt đầu build

B4: Sau khi build xong, vào git Hub pages, sẽ có link để truy cập trang web

Chú ý:

* Mỗi khi sửa code và up lại, trang web sẽ tự động build lại