

CHỦ BIÊN: TRỊNH THÀNH NAM (TRƯỞNG BAN CHUYÊN MÔN),
HOÀNG MẠNH KHIÊM (BAN CHUYÊN MÔN)

GIÁO TRÌNH GIT & GITHUB CƠ BẢN

Là bản lưu hành nội bộ cho các lớp bổ trợ kiến thức CLB TIN HỌC

Tái bản lần thứ 1

CÂU LẠC BỘ TIN HỌC SFIT
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

MỤC LỤC

MỤC LỤC.....	2
LỜI NÓI ĐẦU	3
I.KHÁI NIỆM	4
1. GIT	4
2. GITHUB.....	4
II. CÀI ĐẶT.....	4
I. GIT.....	4
II.GITHUB	5
III. KẾT NÓI GIT VÀ GITHUB THÔNG QUA GIAO THỨC SSH	5
III. MỘT SỐ LỆNH DÙNG TRONG GIT CƠ BẢN.....	6
I.Một số lệnh cơ bản về folder và tệp	6
II. Các lệnh Git cơ bản Quản lí Code.....	6
1. Lệnh cài đặt cấu hình git	6
2. Lệnh bắt đầu một dự án	7
3. Lệnh thay đổi cục bộ	7
4. Lệnh theo dõi các thay đổi.....	7
5. Lịch sử commit	7
6. File đặc biệt.....	7
7. Phân nhánh.....	7
8. Hợp nhất	7
9. Làm việc từ xa	7
10. Cập nhật dữ liệu (Push).....	8
11. Cập nhật dữ liệu (Pull)	8
12. Hoàn lại các thay đổi.....	8
13. Xóa file	8
I.UP CODE LÊN REPOSITORIES	8
II.CẬP NHẬT DỮ LIỆU TỪ REPOSITORIES	9
III. THÊM HOẶC XÓA DỮ LIỆU VÀ UPDATE DỮ LIỆU	9

CHỦ BIÊN: TRỊNH THÀNH NAM (TRƯỞNG BAN CHUYÊN MÔN),
HOÀNG MẠNH KHIÊM (BAN CHUYÊN MÔN)

LỜI NÓI ĐẦU

Giáo trình Git & Github cơ bản được xuất bản để phục vụ cho các bạn theo học tại các lớp bồi dưỡng kiến thức của Câu lạc bộ Tin học – Đại học Giao thông vận tải Hà Nội. Mục đích của việc xuất bản để giúp các bạn nâng cao kiến thức làm việc nhóm và đặt nền tảng sau này. Giáo trình sẽ hỗ trợ các kiến thức cơ bản nhất về kiến thức git và giới thiệu một kho lưu trữ lớn hay mạng xã hội lớn , nơi học tập cho các sinh viên và developer là Github. Kiến thức trong giáo trình trong quá trình học đại học thì sẽ không được đào tạo bài bản và phải tự tìm hiểu, điều này khá là khó khăn. Tại CLB Tin học giáo trình Git và Github cơ bản được cung cấp cho các thành viên theo học các lớp bồi dưỡng một đặc quyền lớn so với những bạn không theo học các lớp bồi dưỡng ở không được cung cấp các giáo trình ngoài chương trình đại học như thế này.

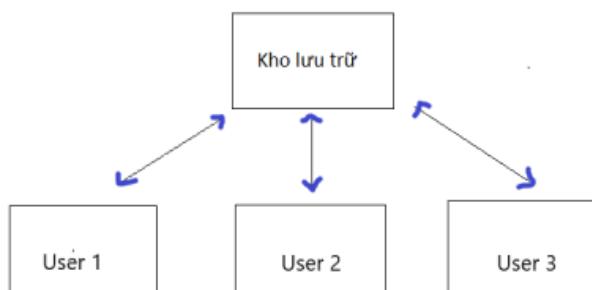
Chúc các bạn học tập thật tốt với giáo trình Git và Github cơ bản của Câu Lạc Bộ Tin Học – SFIT – Trường Đại Học Giao thông vận tải Hà Nội.

BAN CHUYÊN MÔN CLB TIN HỌC

I. KHÁI NIỆM

1. GIT

Git là hệ thống kiểm soát phiên bản phân tán mà nguồn mở (Open Source Distributed Version Control System). Hỗ trợ các dự án thực tế có nhiều nhà phát triển cùng làm việc song song.



2. GITHUB

GitHub là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn Git cho các dự án phần mềm. Github có đầy đủ những tính năng của Git, ngoài ra nó còn bổ sung những tính năng về social để các developer tương tác với nhau.

II. CÀI ĐẶT

I. GIT

Trang web chính thức dành cho Git là <https://git-scm.com/>.

Bước 1: Truy cập trang web và tải về

Bước 2: Nhấp vào logo git và setting theo từng mục



**CHỦ BIÊN: TRỊNH THÀNH NAM (TRƯỞNG BAN CHUYÊN MÔN),
HOÀNG MẠNH KHIÊM (BAN CHUYÊN MÔN)**

Cuối cùng màn hình hiển thị thông báo thành công và đã cài đặt được git thành công

Sau khi cài đặt thành công Git thì Git sẽ tạo một môi trường cho chúng ta thực hiện thông qua cửa sổ điều khiển Git Bash

II.GITHUB

Bước 1: Truy cập: <https://github.com/> để tạo tài khoản github (Để pro nhanh thì có thể follow 2 kênh github của tác giả :

- <https://github.com/hoangmanhkhieu> (Hoàng Mạnh Khiêm)
- <https://github.com/TTNam67> (Trịnh Thành Nam)

Bước 2: Sau khi tạo tài khoản github thành công thì hãy truy cập và thử một số tính năng mà github mang lại

III. KẾT NỐI GIT VÀ GITHUB QUA GIAO THỨC SSH

Bước 1: Kiểm tra thiết bị xem đã có SSH _KEY hay chưa.

- Mở GitBash trên Window (Terminal cho Mac và Linux)
- “ls -al ~/.ssh”: để check xem trên máy đã có SSH key chưa. Lệnh này sẽ liệt kê hết các file trong thư mục .ssh (nếu tồn tại), nếu không tồn tại thì báo lỗi
- Mặc định, file chứa public key trên Github thường có tên:
 - o id_rsa.pub
 - o id_ecdsa.pub
 - o id_ed25519.pub
- Nếu có rồi thì bỏ qua bước 2 và sang bước 3

Bước 2: Ở bước 1, nếu thiết bị chưa có ssh key thì làm theo cách sau để tạo ssh key cho máy

Kiểm tra OpenSSH Client được cài đặt chưa. Thông qua Setting

1. Mở panel Settings, nhấp vào Apps.
2. Dưới heading Apps and Features, click vào Optional Features.

Lướt xuống list để tìm xem OpenSSH Client được liệt kê chưa

Nếu chưa, click vào dấu + bên cạnh Add a feature.

Trượt xuống dưới danh sách để tìm và chọn OpenSSH Client.

Cuối cùng, nhấp vào Install.

- Mở PowerShell và nhập: **ssh-keygen** để tạo keyssh trên thiết bị
- **Nên dùng lệnh ssh-keygen -t ed25519 -b 2048 để tạo key**
- Thông thường, public key sẽ được xác định bằng phần mở rộng .pub. Có thể sử dụng Notepad để xem nội dung của private và public key.

Bước 3: [Thêm 1 SSH Key mới vào tài khoản Github](#)

- Copy nội dung từ file id_ed25519.pub vào clipboard
cat ~/.ssh/id_ed25519.pub | clip
- Chọn Settings

- Tại Access section, chọn **SSH and GPG keys**.
- Tại Title field, chọn **label** cho new Key
- Chọn type của key: authentication hoặc signing (đọc thêm: [About commit signature verification](#))
- Trong Key field, dán public vào đó
- Cuối cùng nhấn Add SSH key

III. MỘT SỐ LỆNH DÙNG TRONG GIT CƠ BẢN

I. Một số lệnh cơ bản về folder và tệp

- Truy cập tới một đường dẫn Path, một vị trí cụ thể: cd <Path>
- Tạo thư mục: mkdir <NAME>
- Xóa thư mục rỗng: rmdir <NAME>
- Xóa thư mục/ tệp: rm -rf <NAME>
- Hiển thị các phần tử con trong 1 thư mục: ls
- Hiển thị các phần tử con trong 1 thư mục rỗng: ls -d
- Hiển thị các phần tử con trong thư mục với các kích thước: ls -s
- Hiển thị các phần tử con trong thư mục với thông tin chi tiết: ls -l
- Hiển thị các phần tử con trong thư mục có thể đọc được: ls -lh
- Hiển thị các tệp ẩn: ls -a
- Hiển thị tất cả các tệp (bao gồm tệp ẩn) với thông tin chi tiết: ls -la
- Tạo tệp mới: touch <NAME>
- Mở VS CODE:
 - Mở VS CODE với dữ liệu tại vị trí Path hiện tại: code * hoặc code .
 - Mở VS CODE với dữ liệu tại Path bất kì: code <Name>
- Copy một hay nhiều tệp đến 1 thư mục mới: cp <Name>
- Di chuyển một hay nhiều tệp đến một thư mục mới: mv <Name>
- Xem nội dung của tập tin: cat <Name>
- Làm sạch cửa sổ lệnh: clear
- Hiển thị một nội dung nào đó : echo

II. Các lệnh Git cơ bản Quản lí Code

1. Lệnh cài đặt cấu hình git

- Đặt email: Git config --global user.email <EMAIL>
- Đặt tên: Git config --global user.name <Name>
- Kiểm tra cài đặt cấu hình: git config -list

2. Lệnh bắt đầu một dự án

- Git init: Tạo một local repository (Kho lưu trữ cục bộ)
- Git clone: Tạo một bản sao cục bộ của kho lưu trữ từ xa (remote repository)

3. Lệnh thay đổi cục bộ

- Git add: Thêm file vào staging area (chỉ mục index)
- Git add <name>: Thêm file name vào staging area
- Git add * hoặc add .: Thêm tất cả các file vào staging area
- Git commit -m <text>: Ghi lại hoặc chụp lại toàn bộ file kèm một tin nhắn

4. Lệnh theo dõi các thay đổi

- Git diff : Theo dõi các thay đổi chưa được dàn dựng (stage) (vẫn ở trong staging area chưa up lên lưu trữ máy chủ)
- Git diff -staged: Theo dõi các thay đổi khi đã commit thành công
- Git status: Hiển thị trạng thái của thư mục làm việc và staging area

5. Lịch sử commit

- Git log: Hiển thị các commit gần đây nhất và trạng thái của phần đầu
- git log -oneline: Hiển thị đầu ra dưới dạng một commit trên mỗi dòng:
- git log --stat: Hiển thị các file đã được sửa đổi
- git log -p: Hiển thị các file đã sửa đổi với vị trí
- git blu <tên file> : Hiển thị sửa đổi từ commit gần nhất trên mỗi dòng của file

6. File đặc biệt

- .gitignore: Chỉ định các file không được theo dõi mình muốn mà Git nên bỏ qua.
- touch .gitignore: Tạo .gitignore
- git ls-files -i --exclude-standard: Liệt kê các file bị bỏ qua
- .gitattribute

7. Phân nhánh

- Git branch <branch name>: Tạo nhánh
- Git branch -list: Hiển thị các nhánh
- Git branch -d <branch name>: Xóa nhánh ở local repo
- Git push origin -delete <branch name>: Xóa nhánh ở remote (github, gitlab,...) repo
- Git branch -m <current branch name> <new branch name>: Đổi tên nhánh
- (Git switch và git checkout)
- Git switch: Chuyển đổi giữa các nhánh
- Git checkout: Chuyển đổi các nhánh trong kho lưu trữ

8. Hợp nhất

- Git merge: Hợp nhất các nhánh
- Git rebase: Áp dụng một chuỗi các commit từ các nhánh khác nhau thành một commit cuối cùng
- git rebase -continue: Tiếp tục quá trình khôi phục
- git rebase -skip: Hủy quá trình khôi phục
- git rebase -i: Thao tác trên các commit hiện có như sửa, sắp xếp,...

9. Làm việc từ xa

- Git remote add: Thêm từ xa cho kho lưu trữ
- Git remote -v: Kiểm tra cấu hình máy chủ từ xa

**CHỦ BIÊN: TRỊNH THÀNH NAM (TRƯỞNG BAN CHUYÊN MÔN),
HOÀNG MẠNH KHIÊM (BAN CHUYÊN MÔN)**

- Git fetch: Tìm nạp dữ liệu máy chủ từ xa
- Git remote rm: Xóa kết nối từ xa
- git remote show: hiển thị một số thông tin cụ thể
- git push origin main: Đẩy dữ liệu đến máy chủ từ xa
- git pull origin main: Lấy dữ liệu từ máy chủ từ xa

10. Cập nhật dữ liệu (Push)

- Git push: Chuyển các commit từ kho lưu trữ cục bộ đến kho lưu trữ máy chủ
- Git push origin main: Đẩy dữ liệu từ kho lưu trữ cục bộ đến kho lưu trữ máy chủ
- Git push -f: buộc đẩy dữ liệu
- Git push origin -delete edited : Xóa nhánh bằng lệnh đẩy

11. Cập nhật dữ liệu (Pull)

- Git pull origin main: Kéo lấy dữ liệu từ kho lưu trữ cục bộ
- Git pull: lấy một nhánh từ xa
- Git fetch <url> : tải xuống các nhánh và thẻ từ một hoặc nhiều kho lưu trữ máy chủ
- Git fetch: Tìm nạp một nhánh cụ thể
- Git fetch -all: tìm nạp tất cả các nhánh
- Git fetch origin: Đồng bộ hóa kho lưu trữ cục bộ

12. Hoàn lại các thay đổi

- git revert: Hoàn lại các thay đổi hoặc một commit cụ thể
- git reset -<hard/soft/mixed>: Đặt lại các thay đổi

13. Xóa file

- git rm <name>: Xóa file khỏi cây làm việc và khỏi chỉ mục
- git rm -cached: Xóa file khỏi git nhưng giữ file trong kho lưu trữ cục bộ của người dùng

III.UP CODE LÊN REPOSITORIES

- Bước 1: Tạo một repositories trên github
- Bước 2: Copy địa chỉ repositories đó
- Bước 3: Mở Git Base: git clone <link>
- Bước 4: cd đến folder đó, để thiết lập code và git
- Bước 5: Gõ git init để tạo một kho lưu trữ cục bộ tại cái folder mìng vừa git về
- Bước 6: Làm việc trên khu vực đó được cái file hoặc folder, ta sẽ dùng lệnh git add * để đưa toàn bộ vào vùng đàm dụng
- Bước 7: Dùng git commit để thêm thông tin thay đổi cho tất cả các file nằm trong vùng đàm dụng
- Bước 8: Gõ lệnh git push origin main để up code lên kho lưu trữ máy chủ là Github

IV. CẬP NHẬT DỮ LIỆU TỪ REPOSITORIES

- Trường hợp xảy ra khi dự án tại máy chủ đã được những nhánh khác push lên thì khi ấy dữ liệu trên máy chủ với kho lưu trữ cục bộ không được đồng bộ. Khi đó ta phải cập nhật dữ liệu từ máy chủ về kho lưu trữ cục bộ.
- Bước 1 : Truy cập folder quản lý các dữ liệu ở kho lưu trữ cục bộ
- Bước 2: Dùng lệnh git pull origin main để đồng bộ toàn bộ dữ liệu từ kho lưu trữ máy chủ về kho lưu trữ cục bộ

V. THÊM HOẶC XÓA DỮ LIỆU VÀ UPDATE DỮ LIỆU

- Thêm hoặc xóa tùy theo mục đích của người sử dụng
- Sau khi có bước đó thì chúng ta sẽ làm theo 1 quy trình cố định
- Git add * → git commit → git push.

CHỦ BIÊN: TRỊNH THÀNH NAM (TRƯỞNG BAN CHUYÊN MÔN),
HOÀNG MẠNH KHIÊM (BAN CHUYÊN MÔN)

**BẢN QUYỀN THUỘC VỀ CLB TIN HỌC SFIT – TRƯỜNG ĐẠI HỌC
GIAO THÔNG VẬN TẢI**

Là bản lưu hành nội bộ trong các lớp học bổ trợ
Tái bản lần thứ 1

Facebook: <https://www.facebook.com/sfit.utc>
Email: sfit.utc@gmail.com