

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



MÔN HỌC: KỸ NĂNG NGHỀ NGHIỆP

ĐỀ TÀI: LÀM QUEN VỚI GITHUB

GV hướng dẫn: Thái Huy Tân

Nhóm sinh viên thực hiện:

<i>Họ và tên</i>	<i>MSSV</i>
Võ Thanh Nhàn	22521009
Lê Tiến Quyết	21520428
Nguyễn Viết Đức	22520273
Huỳnh Nhật Minh	22520862
Bùi Nhật Phi	22521078

Mục lục

1	Giới thiệu về Git, GitHub và giao diện cơ bản	2
1.1	Giới thiệu về Git	2
1.2	Giới thiệu về GitHub	3
1.2.1	Repository	3
1.2.2	Commit	4
1.2.3	Branch	4
1.3	Giao diện cơ bản	4
1.3.1	<> Code	5
1.3.2	Issues	5
1.3.3	Pull request	5
1.3.4	Project	6
1.3.5	Wiki	6
1.3.6	Security	6
1.3.7	Insight	6
2	Các tập lệnh cơ bản	7
2.1	git init	7
2.2	git clone	7
2.3	git branch	7
2.4	git add	8
2.5	git commit	8
2.6	git push	8
2.7	git remote	8
2.8	git log	8
2.9	git pull	9
2.10	git checkout	9
2.11	git merge	9
3	Tổng kết	10

Chương 1

Giới thiệu về Git, GitHub và giao diện cơ bản

1.1 Giới thiệu về Git

Git là hệ thống kiểm soát mã nguồn mở và miễn phí.

Git ra đời vào năm 2005 và được phát triển bởi [Linus Torvalds](#) để hỗ trợ viết Linux kernel.



Hình 1.1: Hình ảnh của Linus Torvalds

Toàn bộ code và lịch sử thay đổi dữ liệu sẽ được lưu trữ trên máy người dùng. Lợi ích khi sử dụng Git:

- Dễ sử dụng, thao tác nhanh, gọn, lẹ và rất an toàn.
- Giúp quy trình làm việc nhóm đơn giản hơn.
- Có thể giúp mọi người làm việc mọi lúc mọi nơi.

1.2 Giới thiệu về GitHub

Sự phát triển của nền tảng bắt đầu vào ngày 19 tháng 10 năm 2007. Trang web được đưa ra vào tháng 4 năm 2008 do [Tom Preston-Werner](#), [Chris Wanstrath](#) và [PJ Hyett](#) thực hiện.



Hình 1.2: PJ Hyett, Tom Preston-Werner, Chris Wanstrath
(lần lượt từ trái sang phải).

Được xem như mạng xã hội của lập trình viên. Giúp mọi người có thể cùng giao tiếp với nhau, cùng nhau thảo luận, cùng nhau làm việc.

Có 3 khái niệm quan trọng: Repository, Commit và Branch.

1.2.1 Repository

Repository là một kho lưu trữ các files của dự án. Các files đó có thể là: code, hình ảnh, âm thanh ... mọi thứ liên quan đến dự án. Hai loại kho lưu trữ trong Github là Local Repository và Remote Repository.

- Local Repository : là một Repository nằm trên máy tính của bạn, có thể đồng bộ hóa với Remote Repository bằng các câu lệnh của git.

- Remote Repository : là một Repository được cài đặt trên server chuyên dụng như là GitHub, GitLab ...

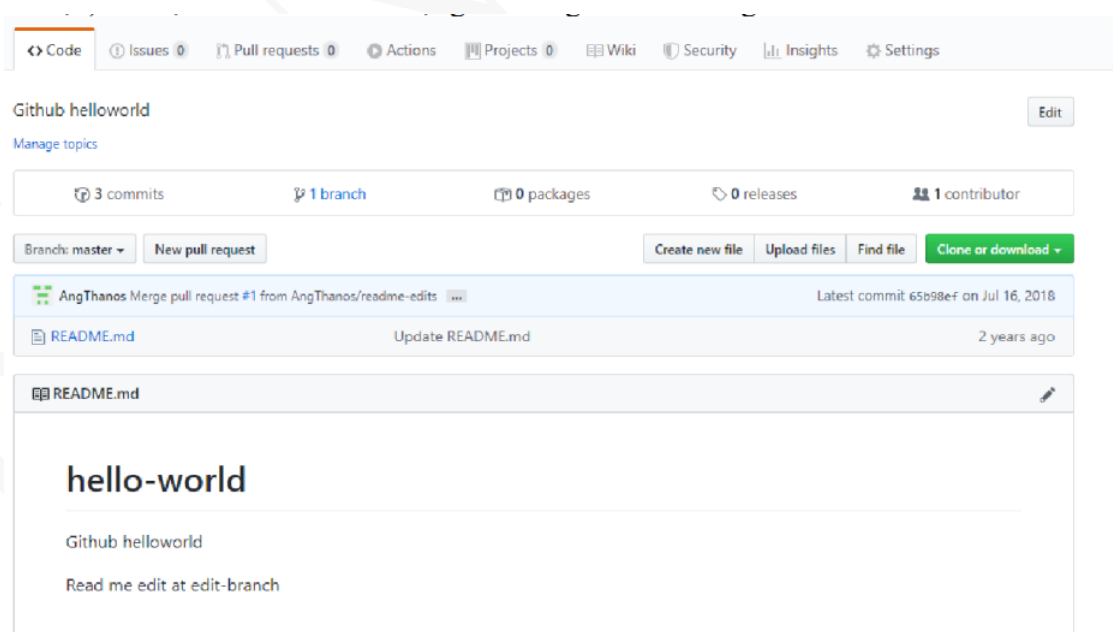
1.2.2 Commit

Là những hành động thêm/xóa/sửa file hay thư mục vào trong Repository. Repository lúc này sẽ tạo ra commit để ghi lại sự khác biệt từ trạng thái commit lần trước với trạng thái hiện tại. Các commit được chứa tại Repository nối tiếp nhau theo thứ tự thời gian.

1.2.3 Branch

Branch được dùng để phân nhánh và ghi luồng của lịch sử. Dùng branch để triển khai dự án theo hướng cô lập để không ảnh hưởng đến công việc của những người khác. Khi công việc đã hoàn thành, bạn có thể nhập branch của mình vào branch cũng những người khác.

1.3 Giao diện cơ bản



Hình 1.3: Giao diện sử dụng của một project

1.3.1 <> Code

Tổng quan :

- Nơi để xem các tệp và thư mục trong Repository.
- Biết được cấu trúc file của dự án.
- Xem qua từng file và xem được lịch sử commit.

Branch :

- Liệt kê tất cả các branch có trong Repository.
- Có thể chọn để chuyển giữa các branch với nhau.

1.3.2 Issues

Tổng quan :

- Nơi để quản lý và theo dõi các task liên quan đến dự án, bug, yêu cầu features.
- Các issues có thể được gán nhãn (labels), giao cho từng cá nhân và gắn với pull request.

Labels : quản lý các nhãn gắn dán vào issues.

Milestone : nhóm các issues lại và có một cách để keep track và kiểm tra quá trình.

New issues : tạo issues mới (dán nhãn, assign người làm).

1.3.3 Pull request

Tổng quan :

- Tab này được dùng để quản lý và xem các Pull request.
- So sánh các branch với nhau.
- Xem được các request đã open và close.

Files changed : khi chọn một Pull request để coi, thì mình có thể xem được sự thay đổi mà request này đưa tới thông qua trang này.

Commit : xem được danh sách cụ thể bao gồm trong request, giúp theo dõi lịch sử thay đổi.

1.3.4 Project

Tổng quan : tab này cho phép tạo ra một project boards để track công việc, sắp xếp task và lên kế hoạch cho các milestone.

1.3.5 Wiki

Tổng quan : là nơi tạo ra và ghi lại doc cho project của mình.

1.3.6 Security

Tổng quan : tab này cung cấp thông tin về bảo mật của Repository bao gồm các lỗ hổng đã biết trong phần phụ thuộc và gợi ý cách khắc phục.

1.3.7 Insight

Tổng quan : cung cấp các phân tích và data liên quan tới Repository. Có thể biết được các thông tin như: code frequency, contributors, ...

Chương 2

Các tập lệnh cơ bản

2.1 git init

Câu lệnh **git init** dùng để tạo một kho lưu trữ trống và mới. Nó tạo một thư mục con có tên là **.git** và thư mục này chứa tất cả các tập tin cần thiết cho kho chứa.

2.2 git clone

git clone + 'URL'

Thực hiện câu lệnh **git clone** để tạo một thư mục mới có tên giống tên của Repository đã tồn tại trên Github, bao gồm tất cả các files, branches và commits vào trong máy.

2.3 git branch

Với cú pháp **git branch** thì dùng để kiểm tra branch hiện tại mà chúng ta đang thực hiện.

Để xem tất cả các branch hiện đang có thì ta dùng câu lệnh:

git branch -r

Ngoài ra để tạo một branch mới thì chúng ta sử dụng câu lệnh:

git branch + 'Name of new branch'

2.4 git add

Sau khi thay đổi dữ liệu ở các files thì bạn cần cập nhật lại các files lên Staging Area với câu lệnh:

git add .

2.5 git commit

Ngay sau khi thực hiện câu lệnh **git add .** ở 2.4 thì cần sử dụng câu lệnh Commit để lưu lại những thay đổi trong Repo:

git commit -m + 'Message about changes'

2.6 git push

Khi đã thực hiện câu lệnh Commit ở 2.5 thì thông tin mới chỉ được cập nhật lên Local Repository.

Để cập nhật lên server thì dùng câu lệnh sau:

git push origin + 'Name of branch'

Muốn xóa một branch nào đó trên server thì ta có thể dùng:

git push origin -delete + 'Name of branch to delete'

2.7 git remote

Ngoài việc cập nhật lên server như đã nói ở 2.6. Nếu chưa tồn tại remote trên server thì ta cần phải thêm mới một remote trước rồi mới push.

Các câu lệnh để thực hiện điều đó là:

git remote add origin 'Remote URL'

git push origin 'Name of branch'

2.8 git log

Sử dụng câu lệnh **git log** sẽ cho chúng ta biết thông tin về *người commit*, *ngày giờ*, *message* của những lần commit đó.

2.9 git pull

Sau một khoảng thời gian nào đó, sẽ có những thay đổi dữ liệu các files ở trên server. Để cập nhật những thay đổi đó về branch hiện tại trên máy thì chúng ta cần sử dụng câu lệnh:

git pull origin + 'Name of branch'

2.10 git checkout

Để làm việc trên một nhánh nào đó thì chúng ta cần di chuyển tới nhánh đó với câu lệnh:

git checkout + 'Name of branch'

2.11 git merge

Sau khi cập nhật các file và push lên branch mới. Giả sử bây giờ mình cần ghép các files vào nhánh gốc (**main**).

Đầu tiên, chúng ta cần di chuyển tới nhánh gốc với câu lệnh:

git checkout main

Sau đó để gộp branch cần gộp vào nhánh gốc thì chúng ta dùng lệnh **merge**:

git merge + 'Name of branch to merge'

Chương 3

Tổng kết

Trong bài viết này thì nhóm đã giới thiệu về Git, GitHub và một số câu lệnh mà nhóm đã tìm hiểu được. Hy vọng qua bài viết này sẽ giúp các bạn hiểu thêm về Git và cách sử dụng chúng trong công việc sao cho hiệu quả.

Tài liệu tham khảo

- [1] Những lệnh git cơ bản cần nhớ (Nguyen Thi Hang)
- [2] [GIT] Tìm hiểu về Commit và Stagging Area (Thạch Phạm)
- [3] Git Branches: List, Create, Switch to, Merge, Push, Delete (Noble Desktop)
- [4] Từ gà tới pro Git và Github trong 20 phút - Tự học Git siêu tốc (Phạm Huy Hoàng)
- [5] Git (Wikipedia)
- [6] Git Commit là gì? Git Commit được sử dụng như thế nào? (Got It Vietnam)
- [7] Git là gì? Các lệnh git cơ bản mà mọi lập trình viên nên biết (TopDev)
- [8] Github (Wikipedia)
- [9] Github là gì? Repository là gì? Các thuật ngữ liên quan đến Github (TINO GROUP)