**TẤT TẦN TẬT VỀ**

**GIT CƠ BẢN**

**NOTEBOOK của Thug24**

**Cập nhật: 10/1/2023**

**Facebook: Tuấn Hưng**

**Github: gnuh24**

**Lời mở đầu**

Xin chào mọi người, tui là Hưng hiện tại tui chỉ đang là sinh viên bập bẹ tập code mà thôi. Và tui hay có thói quen là với mỗi môn hay là 1 chủ đề gì đó hay thì tui sẽ Note lại trên file Word rồi share cho anh em dĩ nhiên là sẽ không có liên quan tới buôn bán tài liệu gì ở đây. Đây chỉ là Notebook của tui trong quá trình học cho nên nếu có bất cứ sai xót nào thì mong anh em thông cảm và góp ý thông qua mạng xã hội tui để link bên dưới.

Về nội dung của file Notebook này thì chủ yếu về GIT. Các nguồn tham khảo thì mình sẽ để nó ở trang cuối và có đi kèm theo link. Notebook này sẽ tóm tắt lại những kiến thức mà mình thấy nó quan trọng. Notebook này cho anh em biết về các khái niệm về GIT, GITHUB và cách sử dụng. Đồng thời một số hình ảnh minh họa mình có lấy từ khóa học GIT của Udemy.

Facebook: <https://www.facebook.com/profile.php?id=100036421866670>

Github: <https://github.com/gnuh24>

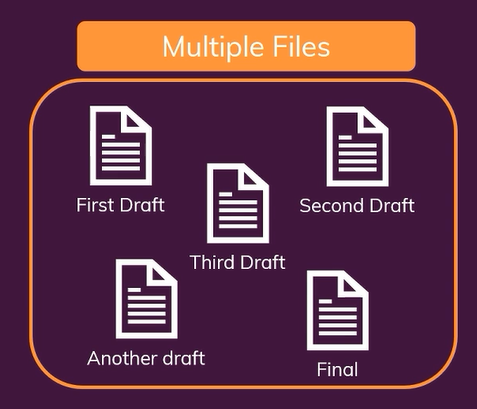
**BÀI 1: GIỚI THIỆU VỀ GIT / GITHUB**

1. **GIT LÀ GÌ ?**

* Git là 1 Version Controll System (VCS) (Hệ thống quản lý các phiên bản.
  1. **Version Management/Controll**
* Đầu tiên giả sử như bạn đang phải code 1 project mà giảng viên giao cho và bạn đang muốn tìm cách quản lý project một cách hiệu quả. Lúc này ta sẽ thường sử dụng theo 2 cách truyền thống là

Multiple files hoặc Single File

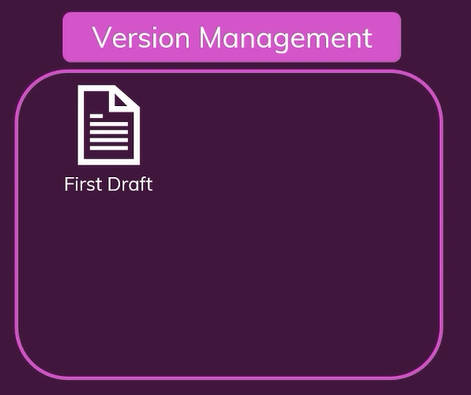
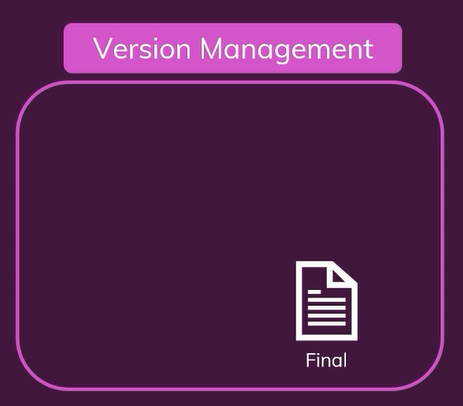
* Giải pháp 1 (Multiple files): Bạn phải code rất nhiều phiên bản khác nhau rồi sau cùng mới cho ra phiên bản cuối cùng của project đó. Lúc này bạn vẫn có thể tìm lại được các phiên bản cũ nếu như final project của bạn bị lỗi gì đó. Tuy nhiên cách làm này sẽ khiến cho folder của bạn bị rối và khá khó chịu trong việc quản lý. Vì vậy ta đi đến giải pháp thứ 2



* Giải pháp 2 (Single File): Lúc này thay vì copy tạo ra nhiều phiên bản khác nhau thì bạn chỉ dùng đúng 1 phiên bản. Lúc này folder của bạn sẽ rất gọn gàng vì chỉ có duy nhất 1 file project nhưng nếu project đó bị hỏng thì coi như ta không có bản back-up nào của phiên bản trước đó để dự trù.



* Sự hạn chế của 2 giải pháp trên khiến ta cần dùng đến giải pháp thử 3 **Version Management/Controll**
* Giải pháp 3 (**Version Management/Controll** ): Kỹ thuật này sẽ phép ta vừa code trên 1 file duy nhất mà không cần phải copy ra, nhưng khác với Single File là ta sẽ giữ được tất cả các phiên bản trước của project và ta có thể back up lại bất cứ lúc nào, bất cứ phiên bản nào mà ta muốn. Và chỉ có phiên bản cụ thể hiện tại của bạn đang dùng là có trên máy của chúng ta. Cụ thể cơ chế này hoạt động thế nào thì sang bài 2 chúng ta sẽ rõ.

**** ****

* **Để thực hiện giải pháp thứ 3 ta cần dùng đến các Version Controll System. Và Git là 1 trong số chúng.**
  1. **Ưu điểm của GIT**
* Là 1 (**VCS**) miễn phí và cục bộ
* Lưu được tất cả lịch sử:

+ Cái gì thay đổi ?

+ Thay đổi khi nào ?

+ Thay đổi ở đâu ?

* Theo dỏi và phát hiện những sự thay đổi của code.
* So sánh được các phiên bản khác nhau của code.
  1. **Nhược điểm của GIT**
* Vì là 1 hệ thống cục bộ cho nên ta chỉ có thể sử dụng nó trên 1 thiết bị.  
  **VD**: Ta có Project A trên máy A thì ta chỉ có thể sử dụng GIT trên máy A để tương tác với project A. Ta không thể dùng máy B (Có GIT) để quản lý project A được.
* Nếu ta bị mất thiết bị thì những dữ liệu cục bộ trong máy đó cũng sẽ bay mất và GIT không giúp ta khôi phục được.
* **Để khắc phục điểm yếu cục bộ này thì ta sẽ cần 1 thứ gì đó có thể đồng bộ và liên kết toàn bộ trong các thiết bị dữ liệu cục bộ lại với nhau.**
* **GITHUB là thứ ta cần tìm.**

1. **GITHUB LÀ GÌ ?**

* Là một dịch vụ lưu trữ đám mây miễn phí
* Là một nền tảng hợp tác và quản lý phát triển các phần mềm.
* Cung cấp 1 kho lưu trữ GIT ngay trên đám mây (K
* Giải quyết được nhược điểm của GIT truyền thống
* Cho phép nhiều người dùng có thể truy cập vào cùng 1 Repository (Kho lưu trữ GIT) để cùng tương tác với 1 dự án.
* Github sử dụng Git để quản lý các repository vì thế mà ta có thể lưu lại lịch sử thay đổi của Repository

**BÀI 2: VERSION MANAGEMENT VỚI GIT**

**GIT HOẠT ĐỘNG NHƯ THẾ NÀO ?**

**TÀI LIỆU THAM KHẢO**

1. Khóa học GIT Udemy: <https://www.udemy.com/course/git-github-practical-guide/>
2. Kênh youtube: <https://www.youtube.com/@TITVvn>
3. Web thực hành: <https://learngitbranching.js.org/?locale=vi>
4. Các lệnh cơ bản: <https://cs.fyi/guide/git-cheatsheet>
5. Software Development: <https://www.atlassian.com/git>