1. **GIT những điều cần biết :**
   1. **Cài đặt Git :** vào google download về kiểm tra cài đặt thành công hay chưa : git - -version
   2. **Cấu hình đầu tiên sử dụng:**
      1. Git config - - global user.name phamanhduc
      2. Git config - -global user.email [phamanhduc536@gmail.com](mailto:phamanhduc536@gmail.com)
   3. **Kiểm tra cấu hình đúng chưa** : git config - - list
   4. **Muốn thay đổi user thì cứ gõ lại như trên**
2. **Local Repository**
   1. **Repository :** 
      1. **Khái niệm : Kho chứa**
         1. Lưu trữ tất cả các trạng thái của tập tin , các hành vi của người dùng trên các tập tin này
         2. Thường bố trí 1 repository tương đương với 1 project mà ta thực hiện
         3. Để ghi lại việc thay đổi trạng thái của tập tin vào repository chúng ta thực hiện thao tác gọi là **Commit**
         4. Khi thực hiện commit trong repository sẽ tạo ra commit(version) đã ghi lại sự khác biệt từ trạng thái đã commit lần trước đến trạng thái hiện tại. Mỗi commit phân biệt nhau bằng 1 id riêng chuỗi kí tự sinh ra có chiều dài 40 kí tự
      2. **Phân biệt**
         1. **Local Repository :** repository lưu trữ trên máy tính cá nhân, 1 người sử dụng
         2. **Remote Repository :** repository lưu trữ trên hệ thống máy chủ chuyên dụng, chia sẻ project cho nhiều người cùng thực hiện
      3. **Các trạng thái của tập tin :**
         1. **Commited** : dữ liệu đã được lưu trữ an toàn trong CSDL
         2. **Modified** : đã thay đổi tập tin nhưng chưa commit vào cơ sở dữ liệu
         3. **Staged** : đánh dấu sẽ commit phiên bản hiện tại của mỗi tập tin đã chỉnh sửa trong lần commit sắp tới
            * **Sinh ra 3 vùng làm việc với Git:**
3. **Git Directory :** nơi Git lưu trữ các siêu dữ kiện và cơ sở dữ liệu cho dự án của bạn
4. **Working Directory :** Các tập tin nằm trong dự án của bạn
5. **Staginf area :** các tập tin sẵn sàng được đưa lên Git directory
   * 1. **Quy trình để commit 1 tập tin**
        1. Working directory được add vào cùng Staged, cùng staged được đẩy vào repository
   1. **Khởi tạo local repo và đưa file readme.txt vào repo**
      1. **Tạo fofder**
      2. **Git init :**  khởi tạo project
      3. **Git status :** kiểm tra trạng thái các file tại repo xem có sự thay đổi nào k , nếu thay đổi và kiểm tra lại sẽ hiện lên sự thay đổi đó
      4. **Git add \* :** để thêm vào vùng sẵn sàng commit (stagin area)
      5. **Git commit –**m “comment”
      6. **Nếu muốn bỏ qua bước git add thì dùng :** Git commit –a –m “commetn”
      7. **Git log :** xem lịch sử commit
   2. **Vòng đời :**
      1. Mỗi tập tin trong thư mục có thể ở 1 trong 2 trạng thái tracked hoặc untracked nghĩa là theo dõi hoặc k theo dõi
   3. **Xóa file khỏi cái vùng stagin (sẵn sàng commit) :**  nghĩa là lỡ đưa vào cái file sai thì xóa nó ra :
      1. git reset HEAD filename Xóa fileName ra khỏi trạng thái Staged (file vẫn còn tồn tại)
      2. git rm --f filename Xóa fileName ra khỏi trạng thái Staged và xóa khỏi ổ cứng (file không còn tồn tại)
   4. **Cấu hình file mà chúng ta k cần theo dõi** 
      1. Touch .gitignore : tạo tập tin
      2. Cập nhập nội dung cho file .gitignore ( có thể k có tên file)
      3. Công cụ online giúp tạo ra các tập tin .gitignore cho các công cụ mới : <https://www.gitignore.io/>
   5. Phục hồi
      1. Quay về trạng thái trước của tập tin git checkout - - ten file
      2. Muốn quay lại 1 thời điểm bất kỳ : Git checkout - - id
   6. Gộp Commit git –amend
   7. Tìm hiểu kỹ về xem lịch sử git log
      1. Git log –p -3 : hiển thị 3 commit gần đây nhất
      2. Git log - - stat : hiển thị commit với các thông tin tóm tắt
6. REMOTE REPOSITORY
   1. Lưu trữ trên hệ thống máy chủ chuyên dụng, chia sẻ project cho nhiều người cùng thực hiện
   2. Thao tác thường gặp:
      1. Clone : sao chép , tải về toàn bộ nội dung remote
      2. Push : đưa dữ liệu từ local lên remote
      3. Pull : cập nhập , đồng nhất dữ liệu từ remote về local
   3. Quan tâm tới link trên git : <https://github.com/PhamAnhuDuc/Sasss-GruntJS.git>
   4. Thao tac clone :
      1. Git clone : tải nó về và được cái tên foder đúng theo ở trên remote
      2. Git clone url filename : tạo tên mới
   5. Xem thông tin remote hiện tại:
      1. Git remote –v : sẽ ra tên ngắn gọn là origin
      2. Git remote show name<tên ngắn gọn>
   6. Cách tạo sự liên kết dữ liệu từ máy cá nhận lên trên remote
      1. Git remote add <name – tên ngắn gọn> <url -repo>
   7. Cách đổi tên : git remote remane <name1><name2>
   8. Cách đẩy dữ liệu lên UPLOAD
      1. Git push <name><branch>
      2. Chú ý : trước khi đẩy những gì chúng ta làm lên thì phải cập nhập từ tk remote trước vì nhỡ phiên bản làm việc khác với nội dung đã làm rồi : Git pull <tên ngắn gọn><nhánh – master>
      3. Trong trường hợp local có những commit , repo có những commit khác nhau thì sẽ gây ra lỗi vì thế dùng cách :
         1. Git pull - - allow-unrelated-history <tên gắn gọn><nhánh>
   9. Phải luôn chú ý : git pull để cập nhập phiên bản mới nhất rồi mới được upload gì thì up