## GIT - GITHUB là gì?

### Git:

* Là một hệ thống kiểm soát phiên bản mã nguồn (VCS) mã nguồn mở, miễn phí được sử dụng rộng rãi để theo dõi thay đổi của các tệp và thư mục trong một dự án.
* Cho phép quản lý lịch sử phát triển của dự án, cộng tác với người khác và dễ dàng quay lại các phiên bản trước đó.
* Hoạt động dựa trên khái niệm kho lưu trữ (repository) chứa tất cả các phiên bản của dự án.

### Github:

* Là một dịch vụ lưu trữ kho lưu trữ Git dựa trên web, cho phép cộng tác và chia sẻ mã nguồn dễ dàng.
* Cung cấp giao diện người dùng trực quan để quản lý kho lưu trữ, xem lịch sử commit, theo dõi thay đổi, tạo pull request và merge code.
* Là nơi tập trung cộng đồng lập trình viên lớn nhất thế giới với vô số dự án mã nguồn mở và mã nguồn riêng.

## **Cách thức hoạt động của Git:**

### Lưu trữ dữ liệu:

* Git lưu trữ dữ liệu dưới dạng các snapshot (ảnh chụp), còn gọi là các trạng thái của mã nguồn tại một điểm thời gian cụ thể.
* Mỗi khi có sự thay đổi trong mã nguồn, Git chỉ lưu trữ những phần thay đổi (diff) thay vì lưu trữ toàn bộ file mới.
* Nhờ vậy, Git giúp tiết kiệm dung lượng lưu trữ và cho phép truy cập nhanh chóng đến các phiên bản trước đó.

### Khu vực làm việc:

* Git chia khu vực làm việc thành 3 phần chính:
  + **Working Tree:** Nơi chứa các tệp mã nguồn đang được chỉnh sửa.
  + **Staging Area (Index):** Nơi lưu trữ các tệp đã được chọn để commit.
  + **Local Repository:** Nơi lưu trữ tất cả các snapshot của dự án.

### Quá trình làm việc:

* **Sửa đổi:** Thay đổi mã nguồn trong Working Tree.
* **Thêm vào Staging Area:** Chọn các tệp đã thay đổi và thêm vào Staging Area bằng lệnh git add.
* **Commit:** Lưu trữ các thay đổi trong Staging Area vào Local Repository bằng lệnh git commit.
* **Push:** Gửi các thay đổi đã commit lên kho lưu trữ remote (trên Github) bằng lệnh git push.

### Các khái niệm quan trọng:

* **Branch:** Nhánh phát triển riêng biệt trong dự án.
* **Merge:** Hợp nhất các thay đổi từ các branch khác nhau.
* **Pull Request:** Yêu cầu hợp nhất code từ một branch khác vào branch chính.

### Ưu điểm của Git:

* Theo dõi thay đổi của mã nguồn dễ dàng.
* Hỗ trợ cộng tác hiệu quả.
* Dễ dàng quay lại các phiên bản trước đó.
* Phân nhánh và hợp nhất code linh hoạt.
* Mã nguồn mở và miễn phí.

## 

## **Các câu lệnh git cơ bản:**

| **Lệnh** | **Tác dụng** | **Ví dụ** |
| --- | --- | --- |
| git init | Khởi tạo kho lưu trữ Git mới | git init my\_project |
| git clone | Sao chép kho lưu trữ Git từ xa | git clone https://github.com/user/repo.git |
| git status | Xem trạng thái các thay đổi trong khu vực làm việc | git status |
| git add | Thêm tệp tin vào khu vực Staging Area | git add index.html |
| git commit | Lưu lại các thay đổi đã được thêm vào Staging Area | git commit -m "Cập nhật giao diện" |
| git push | Gửi các thay đổi đã commit lên kho lưu trữ từ xa | git push origin master |
| git pull | Tải xuống các thay đổi từ kho lưu trữ từ xa và hợp nhất với nhánh hiện tại | git pull origin master |
| git branch | Liệt kê các nhánh hiện có | git branch |
| git checkout | Chuyển đổi sang nhánh khác | git checkout feature/new-feature |
| git merge | Hợp nhất các thay đổi từ một nhánh khác vào nhánh hiện tại | git merge feature/new-feature |
| git reset | Khôi phục các thay đổi | git reset HEAD index.html |
| git stash | Lưu tạm thời các thay đổi | git stash |
| git log | Xem lịch sử commit | git log |
| git config | Thiết lập cấu hình Git | git config --global user.name "Your Name" |

## **Cách thuật ngữ cơ bản của Git:**

| **Thuật ngữ** | **Giải thích** |
| --- | --- |
| Branch | Nhánh, phiên bản cụ thể trong kho lưu trữ tách ra từ project |
| Commit | Thời điểm cụ thể trong lịch sử thực hiện code |
| Check out | Chuyển đổi giữa các branch |
| Fetch | Tải xuống tất cả các bản sao trên hệ thống lưu trữ |
| Fork | Bản sao của một kho lưu trữ |
| Head | Commit mới nhất trong hệ thống lưu trữ |
| Index | Nơi lưu trữ các thay đổi trước khi commit |
| Rebase | Di chuyển các commit |
| Remote | Kho lưu trữ từ xa |
| Master | Nhánh chính trong kho lưu trữ |
| Merge | Bổ sung các thay đổi từ nhánh này sang nhánh khác |
| Origin | Phiên bản đặc biệt của repository, liên lạc với nhánh chính |
| Pull | Yêu cầu thay đổi cho nhánh chính |
| Push | Cập nhật các branch từ xa |
| Repository | Kho lưu trữ Git |
| Stash | Loại bỏ các thay đổi hiện đang có |
| Tags | Theo dõi các commit quan trọng |
| Upstream | Nơi bạn push các thay đổi của mình trong nhánh chính |

## **Một số lưu ý khi dùng Git:**

* **Git Cheat Sheets:** Đây là các website cung cấp Git tự động. Nếu mới sử dụng bạn có thể dùng Git Cheat Sheets để dễ nhớ các lệnh.
* **Commit thường xuyên:** Để tránh lỗi khi kết hợp, bạn nên thực hiện việc commit thường xuyên để dễ dàng tích hợp code với nhau.
* **Test rồi mới commit:** Bạn cần test để hoàn tất quá trình trước khi commit.
* **Viết ghi chú khi commit:** Viết các ghi chú giúp người khác nắm được tiến độ công việc mà bạn đang thực hiện.
* **Thử nghiệm Branch khác:** Sử dụng một nhánh khác để thử nghiệm.
* Theo một Git Workflow: Để giúp cả nhóm hiểu nhau hơn và thực hiện công việc hiệu quả nên sử dụng Git Workflow.