# Tìm hiểu về github

## Github là gì

Github là một dịch vụ cung cấp kho lưu trữ mã nguồn trên nền web cho các dự án phát triển phần mềm.

Dự án trên github cơ thể được truy cập bà thao tác sử dụng một giao diện dòng lênh và làm việc với tất cả các lệnh git tiêu chuẩn

## Github hoạt động như thế nào

### git data transport commands

**Remote repository:** nơi để lưu local repository lên internet nhưng không lưu tất cả mà chỉ lưu nhưng cái được push lên

**Git structures**

Dữ liệu sẽ được lưu trữ tại ba nơi:

* Workspace hay working tree
* local repository
* staging area

Staging Area là khu vực sẽ lưu trữ những thay đổi của bạn trên tập tin để nó có thể commit, vì đối với git muốn commit tập tin nào thì tập tin ấy phải nằm trong Staging Area. Một tập tin khi nằm trong Staging Area thì sẽ có trạng thái là Staged

## Một số lệnh dùng trong github

### Git config

*Dùng để lấy và thiết lập repository hoặc global option*

Thiết lập tên người dùng/ địa chỉ email

|  |
| --- |
| $ git config –global user.name <username>  $git config –global user.email <mailaddress> |

Muốn hiện thị danh sách thiết lập

|  |
| --- |
| $ git config –global –list |

### Git init

*Dùng để khởi tạo repository trống hoặc khởi động một cái hiện có*

|  |
| --- |
| $ git init |

### Git add

*Dùng để thêm nội dung tệp vào chỉ mục. Lệnh này cập nhật chỉ mục bằng cách sử dụng nội dung hiện tại tìm thấy trong working tree. Ta có thể thêm tất cả hoặc thêm một vài phần của của woking tree đó*

|  |
| --- |
| $ git config –global –list |

### Git commit

*Cam kết thay đổi*

|  |
| --- |
| $ git Commit –m “commit message” |

### Git push

*Đẩy các thư mục lên repository*

|  |
| --- |
| $ git push origin <branch> |

### Git clone

*Copy nguyên bản repository trên github.*

|  |
| --- |
| $ git clone nameRepository |

### Git fetch

*Dùng để kiểm tra nội dung của remote repository mà không muốn merge*

### Git stash

*Được sử dụng khi muốn lưu lại các thay đổi****chưa commit****, thường rất hữu dụng khi muốn đổi sang 1 branch khác mà lại đang làm dở ở branch hiện tại.*

|  |
| --- |
| $ git stash |

Muốn xem nội dung của từng thay đổi, ta dùng câu lệnh :

|  |
| --- |
| $ git stash list -p |

Xem nội dung thay đổi cụ thể của lần thứ n

|  |
| --- |
| $ git stash show stash@{n} |

### Git reset

Di chuyển **HEAD** về vị trí commit reset và vẫn giữ nguyên tất cả các thay đổi của file, nhưng loại bỏ các thay đổi ở stage, ta cùng cú pháp sau:

|  |
| --- |
| $ git reset <commit\_id> |

Di chuyển con trỏ **HEAD** về vị tí commit reset và loại bỏ tất cả sự thay đổi của file sau thời điểm commit reset

|  |
| --- |
| $ git reset –hard |

Di chuyển **HEAD** về vị trí commit. Trạng thái của stage và tất cả sự thay đổi của file được giữ nguyên

|  |
| --- |
| $ git reset -soft |

### Git branch

*Dùng để xóa branch*

Git branch –d <branch>

|  |
| --- |
| $ git branch –d <branch> |