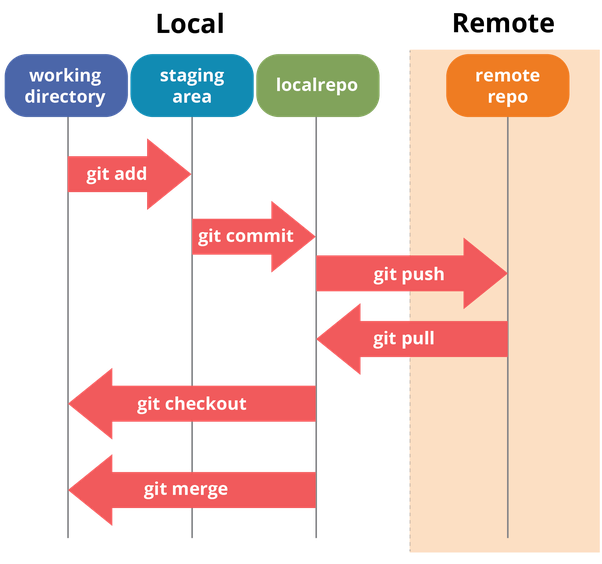
Tutorial Git

# Download and installl Git

[https://git-scm.com](https://git-scm.com/)

# Các vùng làm việc của Git



# Local Repository

## Git init

Sử dụng để khởi tạo **Local repo**

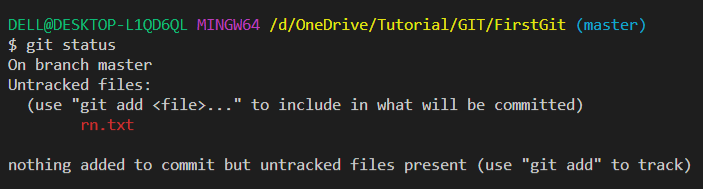
$ git init

Sau khi khởi tạo có thư mục: 

## Git status

**Sử dụng** để xem trạng thái các file ở **Local repo.**

$ git status



## Git add: add file to staged area

**Sử dụng** để thêm tập tin từ vùng **Working directory** vào vùng **Staging area**.

* Thêm một tập tin mới

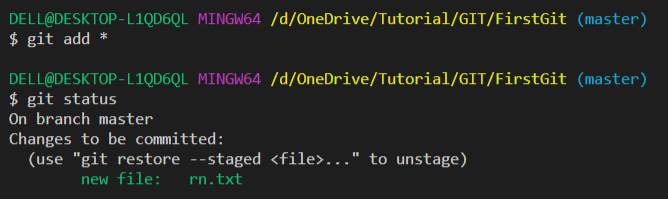
$ git add <file>

Ví dụ: **git add readme.txt**

Thêm tập tin readme.txt mới tạo vào vùng Staging area.

* Thêm nhiều tập tin mới

$ git add .



## Git checkout: restore file

**Sử dụng** để khôi phục file về trạng thái trước khi chỉnh sửa.

$ git checkout --<file>

**Lưu ý**: file checkout phải chưa được sử dụng <**git add**> đưa vào vùng **Staged**. Nếu đã add file hồi phục vào vùng staged thì phải remove file đó ra khỏi vùng này trước khi thực hiện lệnh <**git checkout**>. Để remove file ra khỏi vùng staged, sử dụng lệnh **git restore --staged <file>**.

## Git commit: add file from staged area to local repo

**Sử dụng** để thêm các tập tin đang ở vùng **Staging area** sang vùng **Local repo.**

$ git commit -m “Content commit”

Tham số đi kèm -a: chỉ sử dụng với file đã tạo và commit lên rồi, không sử dụng với file tạo mới chưa commit lên Local respo lần nào.

$ git commit -a – m “Content commit”

Bằng hai câu lệnh:

$ git add .

$ git commit -m “Content commit”

## Git commit –amend: merge with commit before

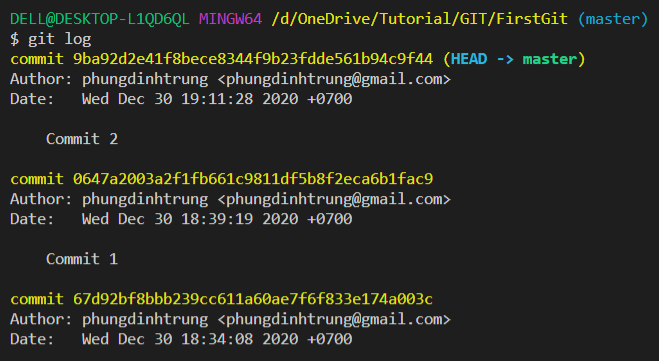
Sử dụng để gộp commit: gộp <commit – amend> với commit trước đó.

$ git commit --amend

## Git log: view list commit

**Sử dụng** để xem các lần đã commit.

$ git log



## Lifecycle

Mô tả vòng đời của file Git. File có hai trạng thái:

* Untracked: là file mới tạo, chưa add, commit lần nào. Chưa được theo dõi (untracked).
* Tracked: là file đã add, commit. Được theo dõi (tracked).
  + unmodified: file đã Add vào vùng staged.
  + modified: file đã edit, chưa Add vào vùng staged
  + staged: file sau khi edit được Add đưa vào staged.

Untracked

Tracked



## Git restore –staged <file>: remove file in staged area

**Sử dụng** để remove file đã add vào vùng staged.

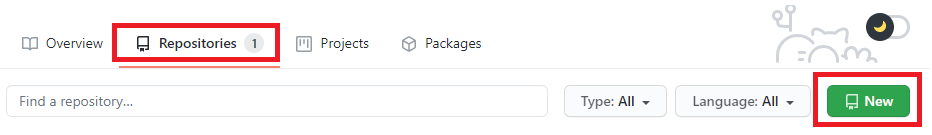
git restore --staged <file>



# Remote repository

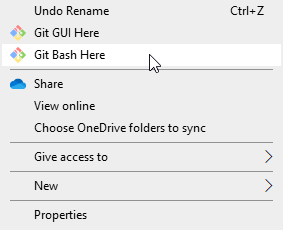
## Tạo remote repository

* Đăng ký tài khoản **GitHub** hoặc **Bitbucket**
* Tạo repository **GitHub**

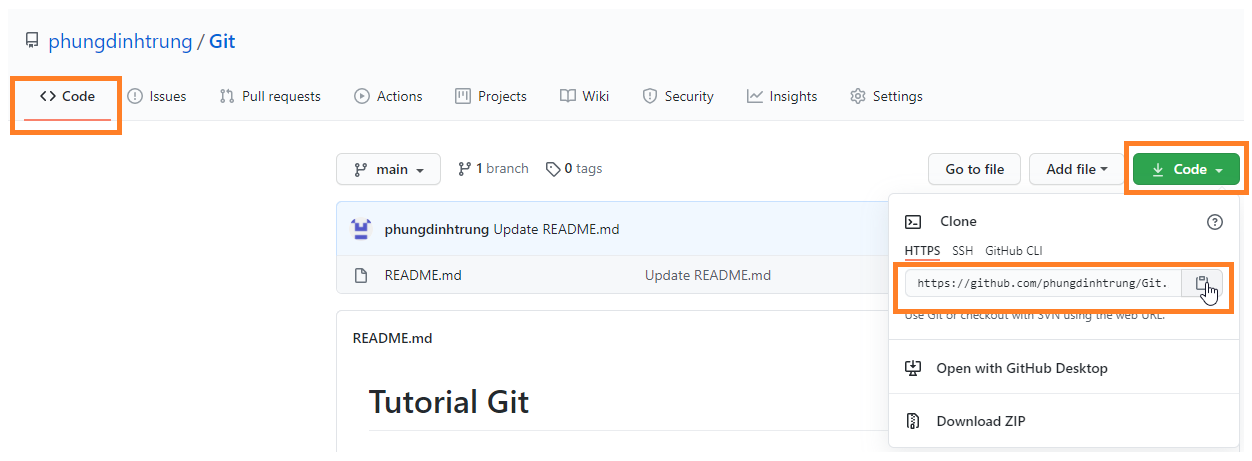


## Git clone

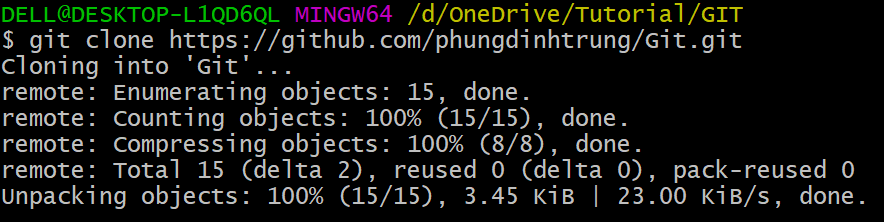
* Tạo folder cần clone dự án Git về
* Nháy chuột phải, chọn Git Bash Here



* Copy đường dẫn https dự án cần tải về



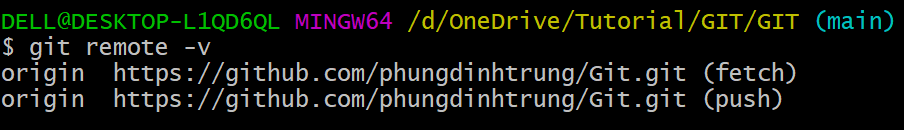
* Đánh lệnh git clone để tải về



## Git remote -v: view info remote

Sử dụng để xem thông tin remote Repo clone về.

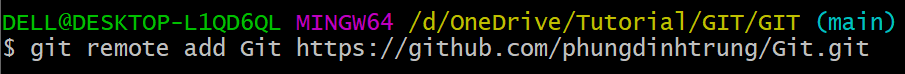
$ git remote -v



## Git remote add <name-remote> <https-remote-repository>: add name-remote

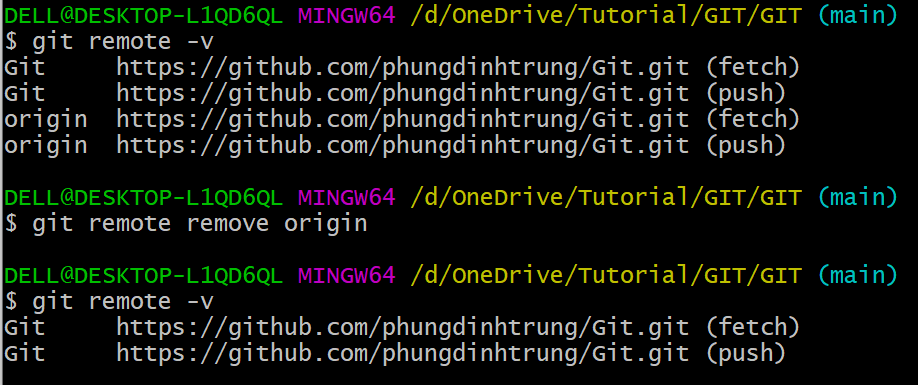
Sử dụng để tạo liên kết Local repo với Remote repo. Liên kết này với tên gốc là **Origin**, có thể đổi lại bằng tên <**name-remote**>.

$ git remote add <name-remote> <https-remote-repository>



## Git remote remove <name-remote>: remove remote link with Local Repo

Sử dụng để loại bỏ link Local repo với Remote repo.

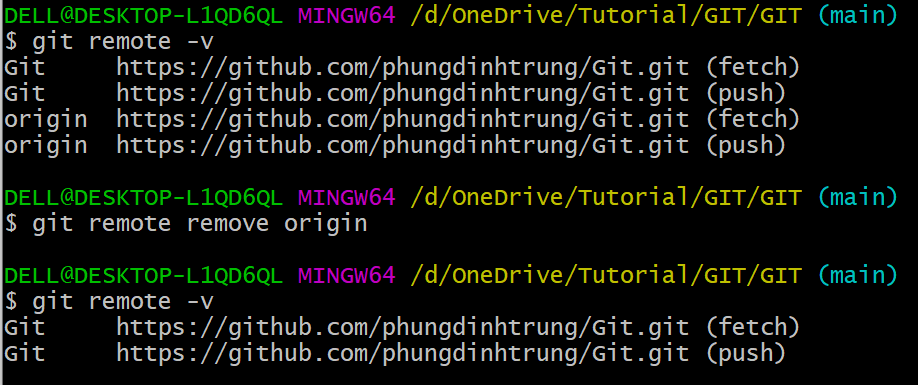
****

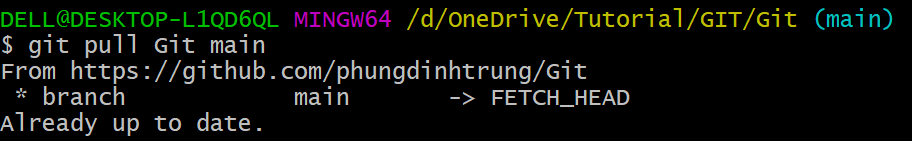
## Git pull <remote-name> <local-name>: get files change from Remote repo

Sử dụng để cập nhật các thay đổi trên **Remote repo** xuống **Local repo**

$ git pull <remote-name> <local-name>

* Để biết Remote name, gõ: **git remote -v**. Hình dưới, remote-name là **Git**.
* Để biết Local name: nhìn vào dấu () cuối đường dẫn ở cửa sổ Git Batch. Ở hình dưới Local name là **main**.

****

Thực hiện cập nhật trên Remote repo xuống Local repo.

## Git push <remote-name> <local-name>: get files change from Remote repo

Sử dụng để cập nhật các thay đổi dưới **Local repo** lên **Remote repo**.

$ git push <remote-name> <local-name>