



Antbuddy.com
Thông Nguyễn
Đạt Phạm

Nội dung:

- Git là gì?
- Các thuật ngữ cơ bản trong git
- Cài đặt git
- Tạo repository.
- Các câu lệnh cơ bản trong git
- Các vấn đề thường gặp và cách xử lý

1. Git là gì?

Git là một trong những Hệ thống Quản lý Phiên bản Phân tán, vốn được phát triển nhằm quản lý mã nguồn (source code) của Linux.

Trên Git, ta có thể lưu trạng thái của file dưới dạng lịch sử cập nhật. Vì thế, có thể đưa file đã chỉnh sửa một lần về trạng thái cũ hay có thể biết được file đã được chỉnh sửa chỗ nào.

```
kbuzdar@sam-VirtualBox:~/gitproject/simplegit-progit$ git log
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master,
origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700

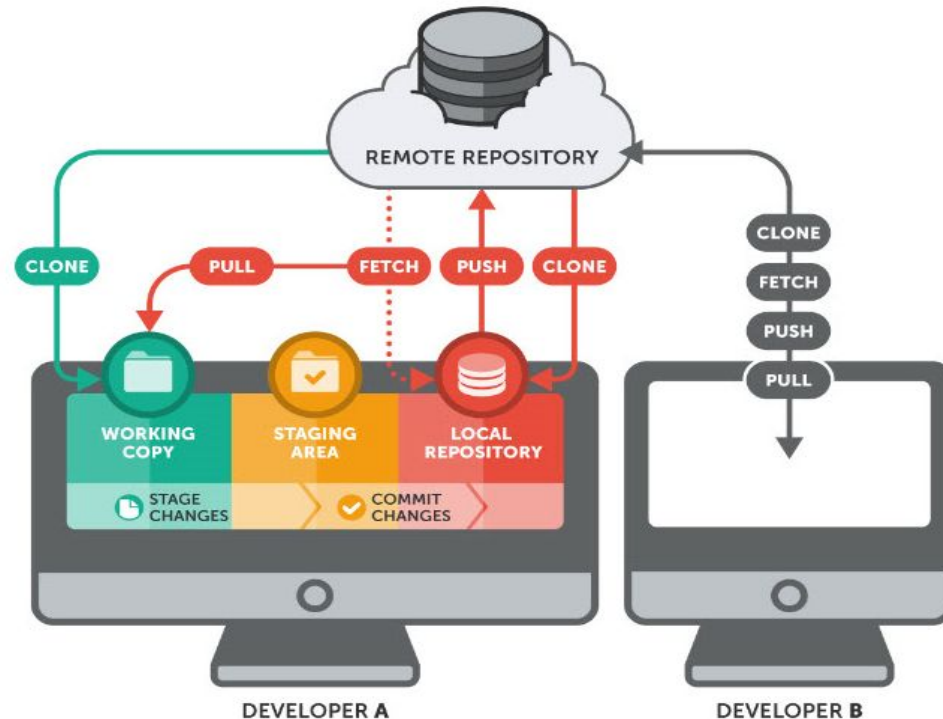
    first commit
kbuzdar@sam-VirtualBox:~/gitproject/simplegit-progit$
```

2. Các thuật ngữ cơ bản trong git

2.1 Repository

Repository được hiểu là một kho lưu trữ nơi chứa các files của dự án. Các file đó có thể là code, hình ảnh, âm thanh hoặc mọi thứ liên quan đến dự án. Bạn có thể tổ chức kho lưu trữ của mình dưới nhiều hình thức khác nhau, hai loại kho lưu trữ trong Github là Local Repository và Remote Repository.

- Local Repository: là một loại repository nằm trên máy tính của bạn, repository này có nhiệm vụ đồng bộ hóa với remote repository bằng các lệnh của git.
- Remote Repository: là một loại repository được cài đặt trên server chuyên dụng. Ví dụ như: GitHub, [GitLab](#), Bitbucket,...



Các thao tác xử lý giữa remote repository và local repository

2. Các thuật ngữ cơ bản trong git

2.2 Commit

- Commit là thao tác báo cho hệ thống biết bạn muốn lưu lại trạng thái hiện hành, ghi nhận lại lịch sử các xử lý như đã thêm, xóa, cập nhật các file hay thư mục nào đó trên repository.
- Khi thực hiện commit, trong repository sẽ ghi lại sự khác biệt từ lần commit trước với trạng thái hiện tại. Các commit ghi nối tiếp với nhau theo thứ tự thời gian do đó chỉ cần theo vết các commit thì có thể biết được lịch sử thay đổi trong quá khứ.
- Khi bạn thực hiện commit, hệ thống đều yêu cầu bạn phải nhập vào commit message để ghi chú tóm tắt là trong lần commit này là bạn đã thực hiện những thay đổi nào, có ý nghĩa thế nào,...



2. Các thuật ngữ cơ bản trong git

2.3 Clone

- Nếu bạn muốn có một bản sao của một kho chứa Git có sẵn, có thể là một dự án mà bạn tham gia - thì bạn hãy thực hiện clone. Đây là điểm khác biệt của Git so với một số hệ thống quản lý phiên bản mã nguồn khác vì clone là tạo ra một bản sao của gần như tất cả những gì của repository mà máy chủ đang lưu trữ. Bạn sẽ có được tất cả lịch sử đã xảy ra trên repository và hoàn toàn có thể quay lại, undo lại từ bất kỳ thời điểm commit nào.

2. Các thuật ngữ cơ bản trong git

2.4 Push

- Lệnh push được sử dụng để đưa nội dung kho lưu trữ cục bộ lên server. Push là cách bạn chuyển giao các commit từ kho lưu trữ cục bộ của bạn lên server.

2. Các thuật ngữ cơ bản trong git

2.5 Fetch

- Lệnh này sẽ truy cập vào repository trên server và kéo toàn bộ dữ liệu mà bạn chưa có từ repository trên server về. Sau đó, bạn có thể tích hợp dữ liệu vào branch bất kỳ lúc nào.

2. Các thuật ngữ cơ bản trong git

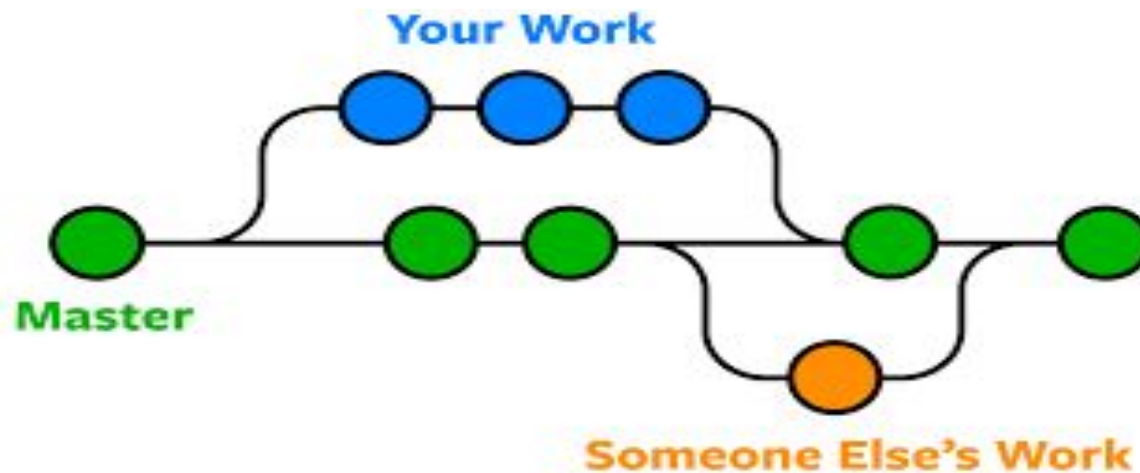
2.6 Pull

- Lệnh này sẽ tự động lấy toàn bộ dữ liệu từ repository trên server và gộp vào cái branch hiện tại bạn đang làm việc.

2. Các thuật ngữ cơ bản trong git

2.7 Branch

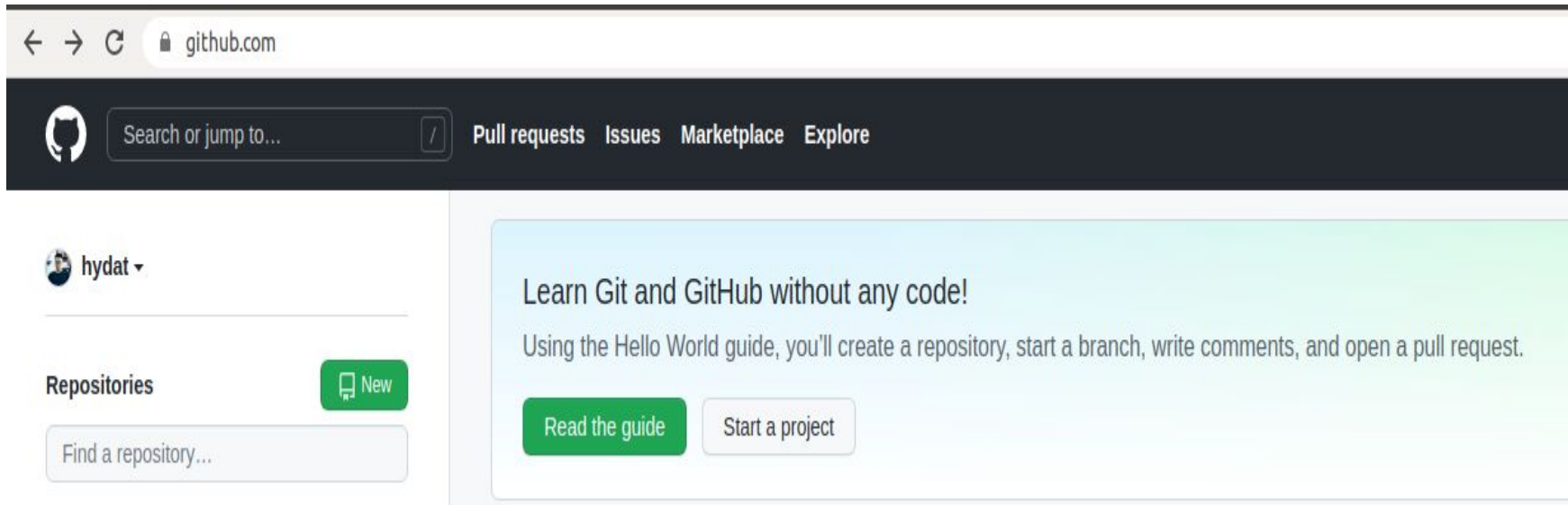
- Các branch (nhánh) đại diện cho các phiên bản cụ thể của một kho lưu trữ tách ra từ project chính của bạn.
- Branch cho phép bạn theo dõi các thay đổi thử nghiệm bạn thực hiện đối với kho lưu trữ và có thể hoàn nguyên về các phiên bản cũ hơn.



3. Cài đặt git

- **MacOS:** <https://git-scm.com/download/mac>
- **Windows:** <https://git-scm.com/download/win>
- **Linux:** <https://git-scm.com/download/linux>

4. Cài đặt github



4. Cài đặt github

Create a new repository

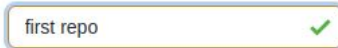
A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *



Repository name *



Great repository name. Your new repository will be created as **first-repo**. How about **automatic-fiesta**?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

4. Cài đặt github

Quick setup — if you've done this kind of thing before

or ☐ HTTPS ☐ SSH

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# first-repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:hydat/first-repo.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:hydat/first-repo.git
git branch -M main
git push -u origin main
```

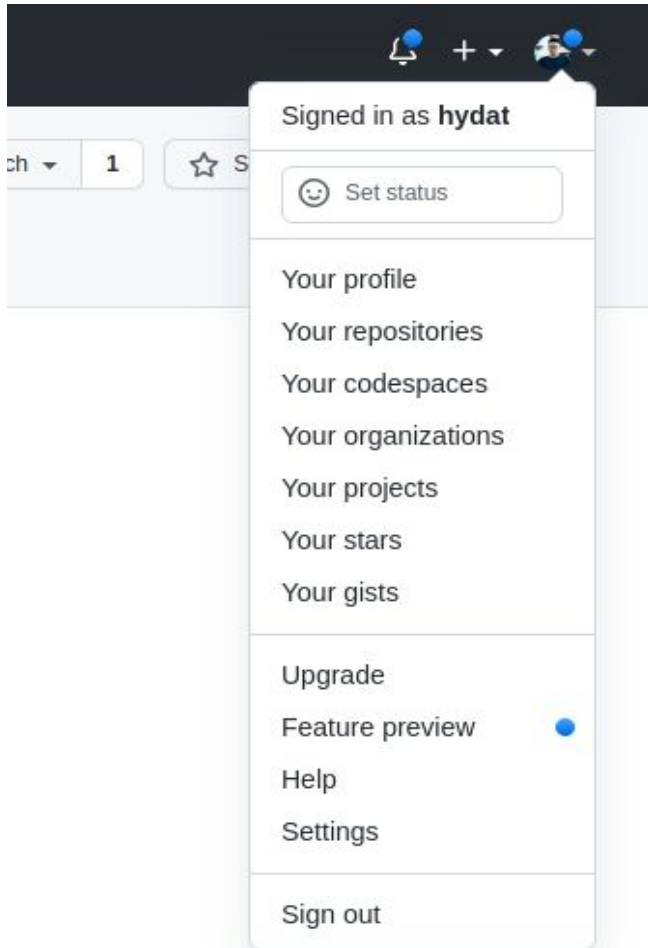
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

 **ProTip!** Use the URL for this page when adding GitHub as a remote.

4. Cài đặt github



4. Cài đặt github



hydat

Your personal account

Go to your personal profile

- Account settings
- Profile
- Account
- Appearance
- Accessibility
- Account security
- Billing & plans
- Security log
- Security & analysis
- Sponsorship log
- Emails
- Notifications
- Scheduled reminders
- SSH and GPG keys
- Repositories
- Packages
- Pages
- Organizations
- Saved replies
- Applications
- Developer settings

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



laptop linux
SHA256:srsVm1IxQpIpASprNVtnNdfFv0rTezEDfmF9PzBwW8w
Added on Jun 2, 2020
Last used within the last week — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode Beta

- ☐ **Flag unsigned commits as unverified**
This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.
[Learn about vigilant mode.](#)

4. Cài đặt github



hydat

Your personal account

[Go to your personal profile](#)

Account settings

Profile

Account

Appearance

Accessibility

Account security

Billing & plans

Security log

Security & analysis

Sponsorship log

Emails

Notifications

Scheduled reminders

SSH and GPG keys

Repositories

Packages

Pages

Organizations

Saved replies

Applications

Developer settings

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

4. Cài đặt github

```
Command Prompt
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\antbu>ssh-keygen -t ed25519 -C "dat.phamtranhy@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\antbu/.ssh/id_ed25519):
Created directory 'C:\Users\antbu/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\antbu/.ssh/id_ed25519.
Your public key has been saved in C:\Users\antbu/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:XDB8ZvMK04B3rNZTl7Xf50R50tqyIbINhSzQ8ZE2oI0 dat.phamtranhy@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|      o++o      o.|
|    =.oB+B . o o|
|  E oo.@.+ . oo|
| . o=o+ . o+|
| ..So.o o =|
| . . . o +.|
| o . + . .|
|   = . +|
| . . .|
+-----[SHA256]-----+

C:\Users\antbu>
C:\Users\antbu>
```

4. Các câu lệnh cơ bản trong git

4.1 git config

- Tác dụng: Để set user name và email của bạn trong main configuration file.
- Cách dùng:
 - Để set tên: `git config --global user.name = "Nguyễn Văn A"`
 - Để set email: `git config --global user.email = "nguyen.a@gmail.com"`

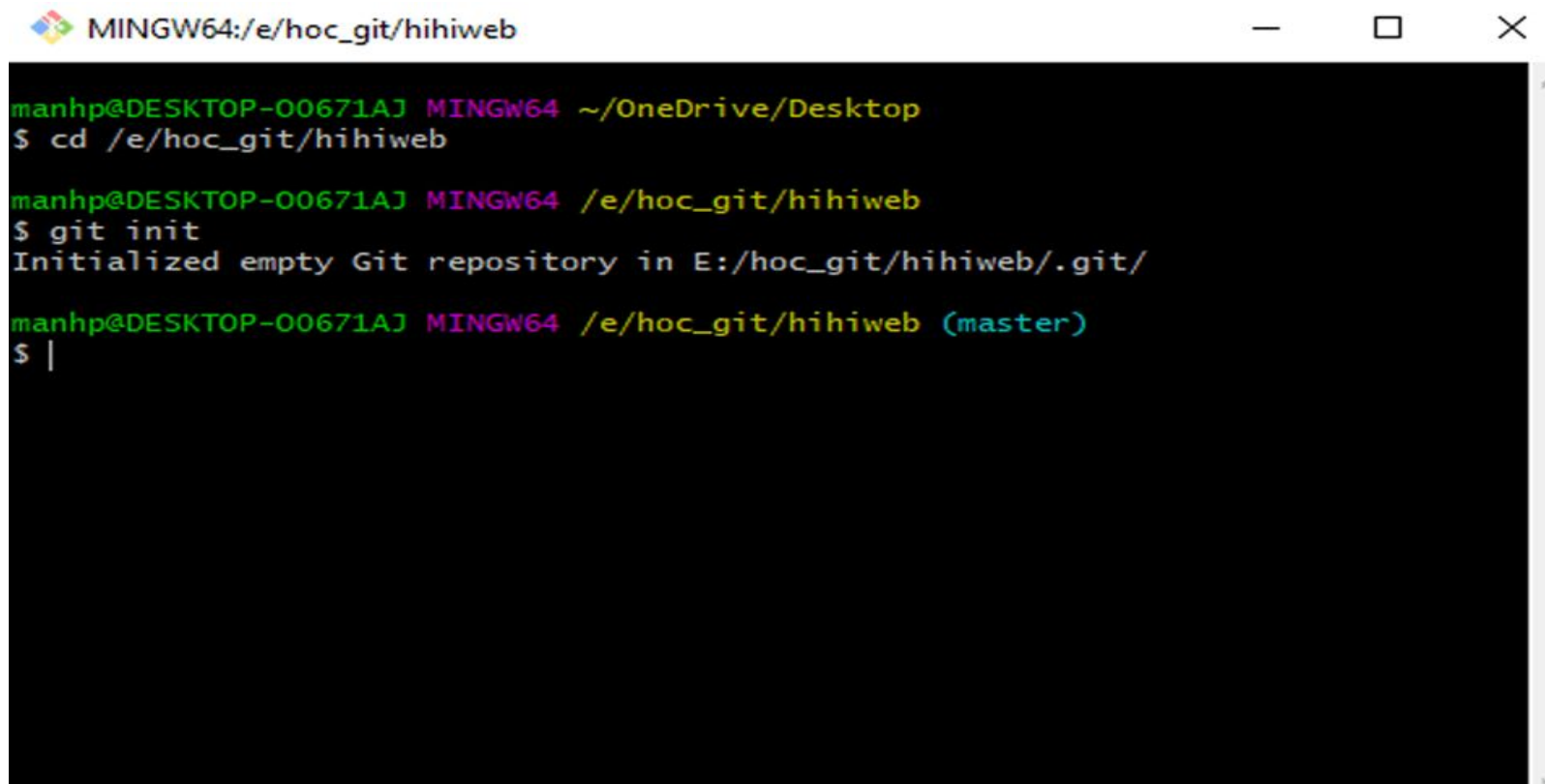
```
root@HowLinux:~# git config --global user.email example@howlinux.com
root@HowLinux:~# git config --global user.name HowLinux

root@HowLinux:~# git config --global -l
user.email=example@howlinux.com
user.name=HowLinux
root@HowLinux:~# █
```

4. Các câu lệnh cơ bản trong git

4.2 git init

- Tác dụng: Khởi tạo 1 git repository 1 project mới hoặc đã có.
- Cách dùng: *git init* trong thư mục gốc của dự án.



```
MINGW64:/e/hoc_git/hihiweb

manhp@DESKTOP-00671AJ MINGW64 ~/OneDrive/Desktop
$ cd /e/hoc_git/hihiweb

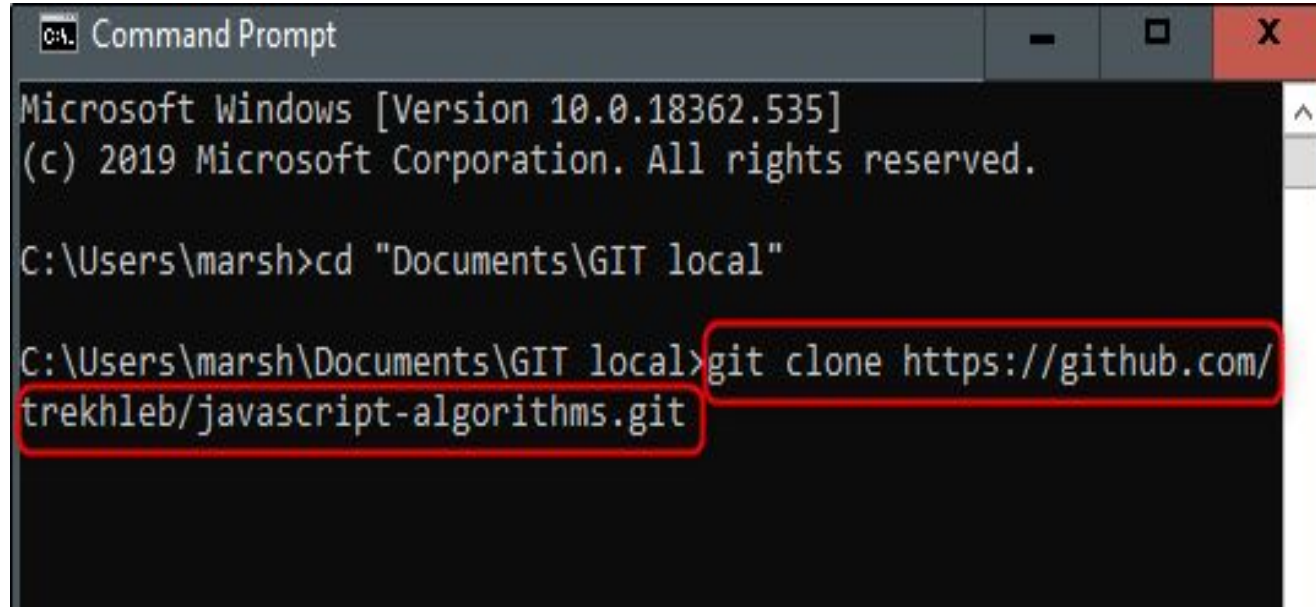
manhp@DESKTOP-00671AJ MINGW64 /e/hoc_git/hihiweb
$ git init
Initialized empty Git repository in E:/hoc_git/hihiweb/.git/

manhp@DESKTOP-00671AJ MINGW64 /e/hoc_git/hihiweb (master)
$ |
```

4. Các câu lệnh cơ bản trong git

4.3 git clone

- Tác dụng: Copy 1 git repository từ remote source.
- Cách dùng: `git clone <clone git url>`.



```
Command Prompt
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.

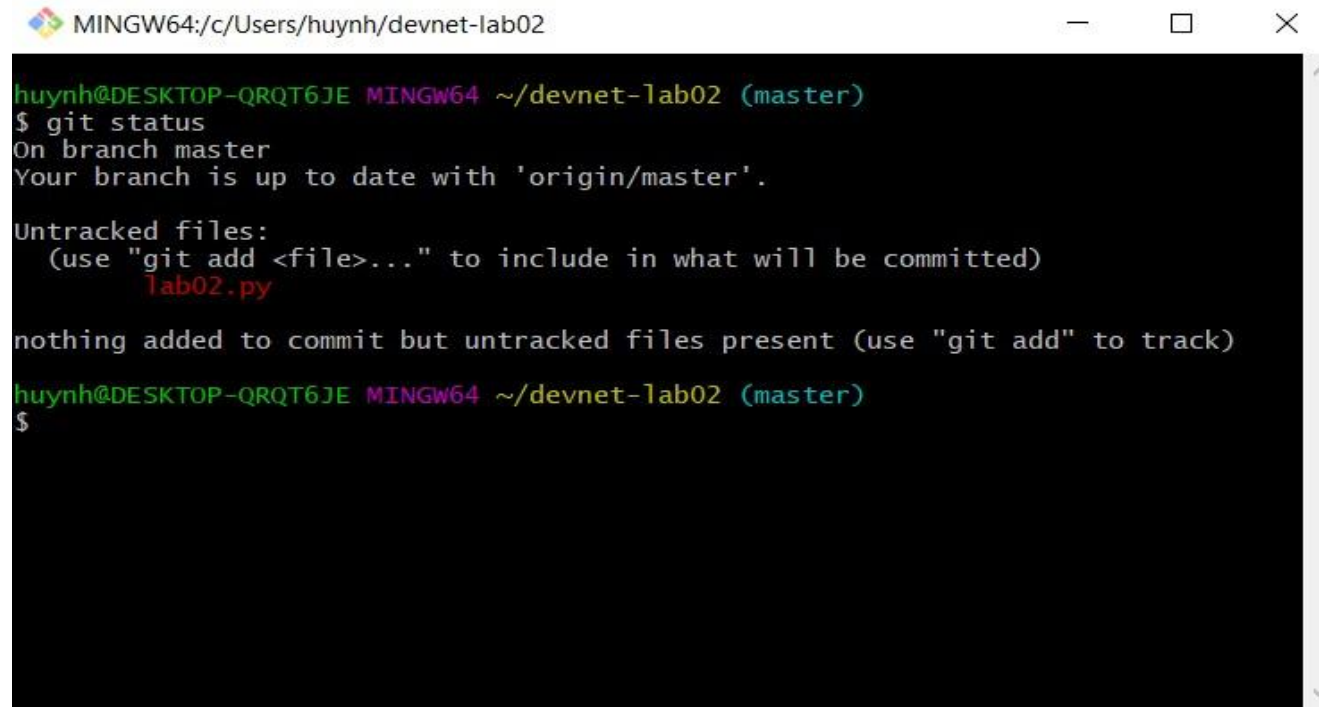
C:\Users\marsh>cd "Documents\GIT local"

C:\Users\marsh\Documents\GIT local>git clone https://github.com/
trekhleb/javascript-algorithms.git
```

4. Các câu lệnh cơ bản trong git

4.4 git status

- Tác dụng: Để check trạng thái của những file bạn đã thay đổi trong thư mục làm việc.
- Cách dùng: `git status` trong thư mục làm việc.



```
MINGW64:/c/Users/huynh/devnet-lab02
huynh@DESKTOP-QRQT6JE MINGW64 ~/devnet-lab02 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

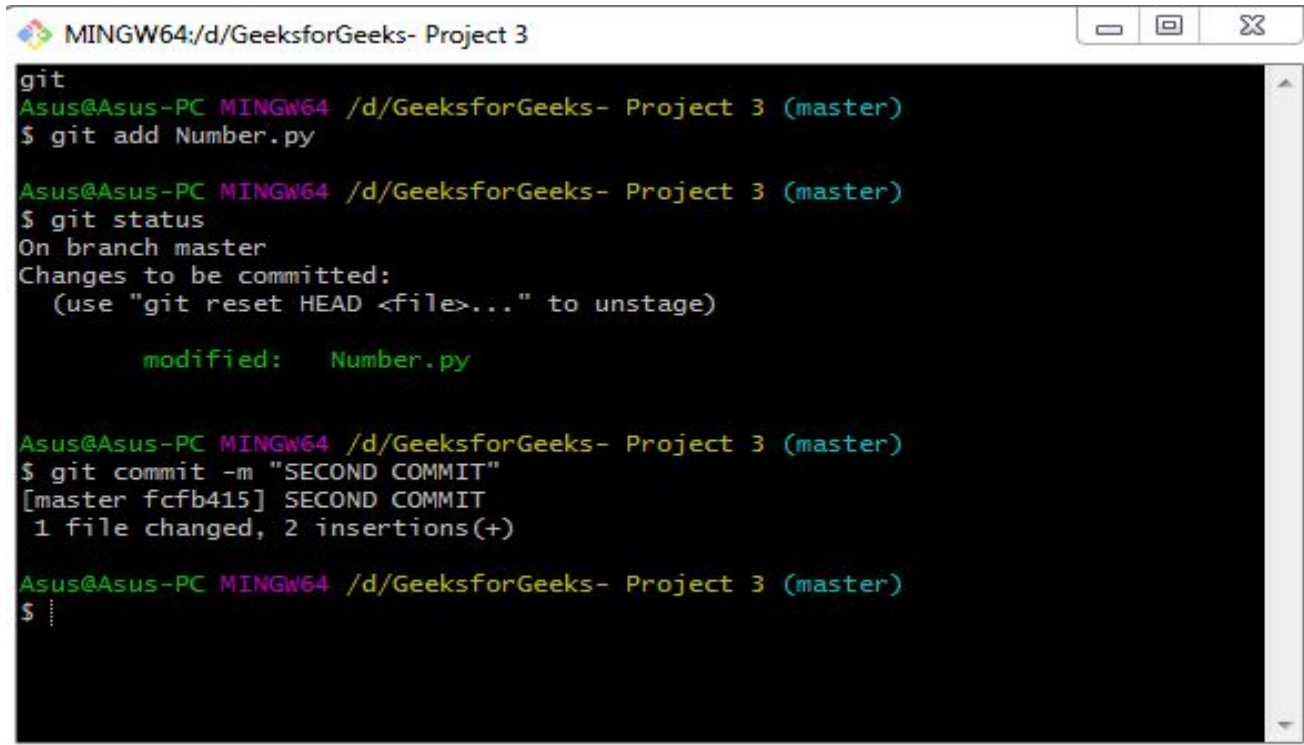
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        lab02.py

nothing added to commit but untracked files present (use "git add" to track)
huynh@DESKTOP-QRQT6JE MINGW64 ~/devnet-lab02 (master)
$
```

4. Các câu lệnh cơ bản trong git

4.5 git add

- Tác dụng: Thêm thay đổi đến stage/index trong thư mục làm việc.
- Cách dùng: *git add <đường dẫn của file thay đổi>*.



```
MINGW64:/d/GeeksforGeeks- Project 3
git
Asus@Asus-PC MINGW64 /d/GeeksforGeeks- Project 3 (master)
$ git add Number.py

Asus@Asus-PC MINGW64 /d/GeeksforGeeks- Project 3 (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   Number.py

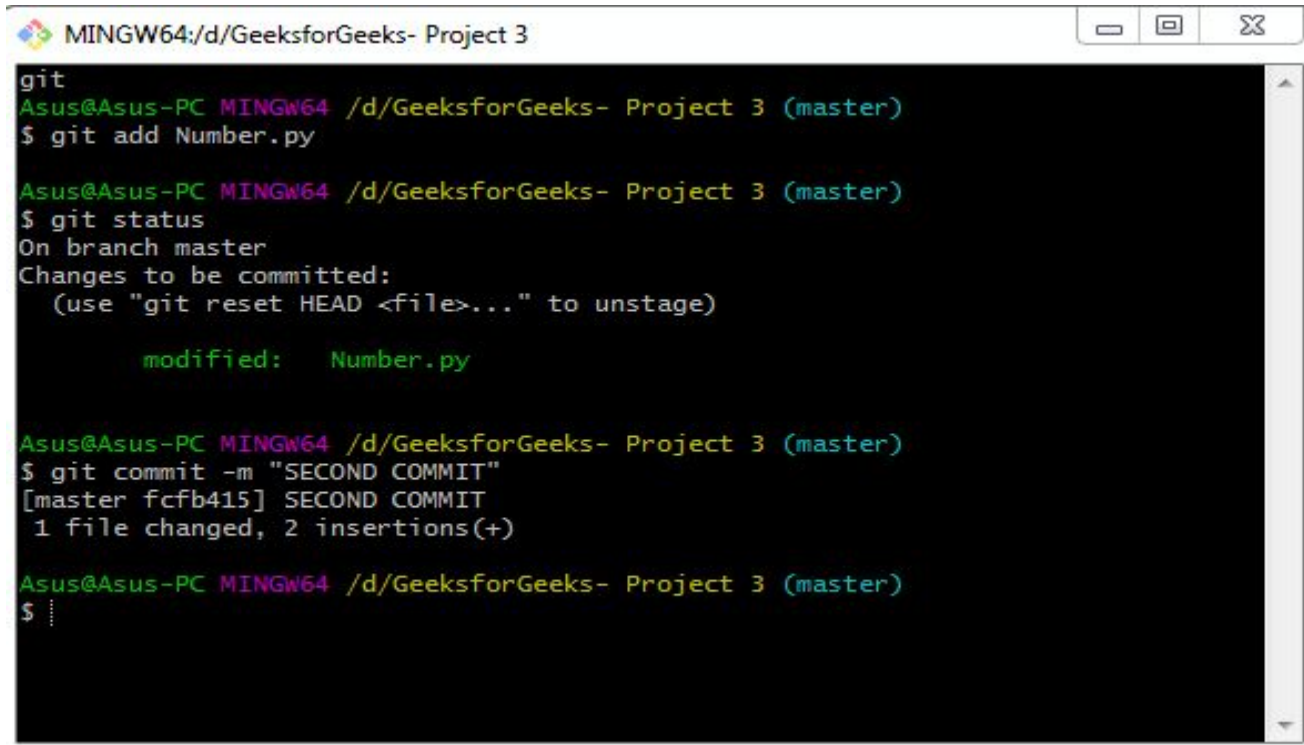
Asus@Asus-PC MINGW64 /d/GeeksforGeeks- Project 3 (master)
$ git commit -m "SECOND COMMIT"
[master fcfb415] SECOND COMMIT
1 file changed, 2 insertions(+)

Asus@Asus-PC MINGW64 /d/GeeksforGeeks- Project 3 (master)
$
```


4. Các câu lệnh cơ bản trong git

4.6 git commit

- Tác dụng: Thêm ghi chú cho mỗi lần thay đổi.
- Cách dùng: `git commit -m <Đây là commit thứ hai>`.



```
MINGW64:/d/GeeksforGeeks- Project 3
git
Asus@Asus-PC MINGW64 /d/GeeksforGeeks- Project 3 (master)
$ git add Number.py

Asus@Asus-PC MINGW64 /d/GeeksforGeeks- Project 3 (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   Number.py

Asus@Asus-PC MINGW64 /d/GeeksforGeeks- Project 3 (master)
$ git commit -m "SECOND COMMIT"
[master fcfb415] SECOND COMMIT
1 file changed, 2 insertions(+)

Asus@Asus-PC MINGW64 /d/GeeksforGeeks- Project 3 (master)
$
```


4. Các câu lệnh cơ bản trong git

4.7 git push

- Tác dụng: Cập nhật những thay đổi từ Local Repository lên Remote Repository.
- Cách dùng: *git push origin <Tên branch>*.

```
lethaianhle@DESKTOP-EK3FF7F MINGW64 ~/Desktop/reactjs-nb (main)
$ git remote -v
origin https://github.com/lethaianhle/reactjs-nb.git (fetch)
origin https://github.com/lethaianhle/reactjs-nb.git (push)

lethaianhle@DESKTOP-EK3FF7F MINGW64 ~/Desktop/reactjs-nb (main)
$ git push origin main
|
```

4. Các câu lệnh cơ bản trong git

4.8 git pull

- Tác dụng: Cập nhật những thay đổi từ Remote Repository về Local Repository.
- Cách dùng: *git pull origin <Tên branch>*.

```
Welcome@Welcome-PC MINGW64 /e/git-demos/pull-tst (master)
$ git pull origin master
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/git-test-jaz/tst-pull-2
 * branch                master       -> FETCH_HEAD
    d24f93c..9f15bb8      master       -> origin/master
Updating d24f93c..9f15bb8
Fast-forward
 1.0_10-Git-branch-file.png | Bin 0 -> 5841 bytes
 1.0_4-Git-bash-option.png  | Bin 0 -> 7389 bytes
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 1.0_10-Git-branch-file.png
 create mode 100644 1.0_4-Git-bash-option.png
```

4. Các câu lệnh cơ bản trong git

4.9 git branch

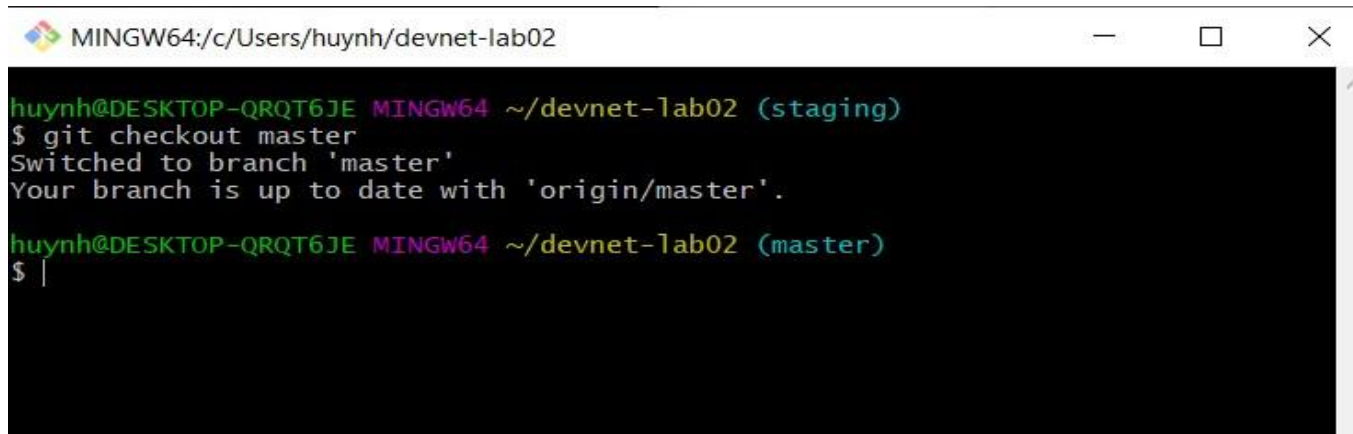
- Tác dụng: Tạo hoặc xem danh sách các branch (nhánh).
- Cách dùng:
 - Xem danh sách các branch: `git branch`
 - Tạo mới một branch: `git branch <Tên branch mới>`

```
[20:42:48] (master) Development
// ♥ git branch
* master
[20:43:54] (master) Development
// ♥ git branch new-branch-name
[20:44:18] (master) Development
// ♥ git branch
* master
  new-branch-name
[20:44:29] (master) Development
// ♥ git checkout new-branch-name
Switched to branch 'new-branch-name'
```

4. Các câu lệnh cơ bản trong git

4.10 git checkout

- Tác dụng: Chuyển sang branch khác.
- Cách dùng:
 - Chuyển sang branch khác: `git checkout <Tên branch>`
 - Tạo mới một branch và chuyển sang branch đó: `git checkout -b <Tên branch mới>`



```
MINGW64:/c/Users/huynh/devnet-lab02
huynh@DESKTOP-QRQT6JE MINGW64 ~/devnet-lab02 (staging)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
huynh@DESKTOP-QRQT6JE MINGW64 ~/devnet-lab02 (master)
$ |
```

4. Các câu lệnh cơ bản trong git

4.11 git merge

- Tác dụng: Gộp nhiều branch thành một branch chung.
- Cách dùng: `git merge <Tên branch muốn merge>`



```
MINGW64:/d/GeeksForGeeks-Branching and merging

Asus@Asus-PC MINGW64 /d/GeeksForGeeks-Branching and merging (master)
$ git branch
* master
  next-five-even-odd

Asus@Asus-PC MINGW64 /d/GeeksForGeeks-Branching and merging (master)
$ git merge next-five-even-odd
Updating a8d9e6b..3c477e1
Fast-forward
 Even-numbers.txt | 10 ++++++++
 Odd-Number.txt   | 10 ++++++++
 2 files changed, 18 insertions(+), 2 deletions(-)

Asus@Asus-PC MINGW64 /d/GeeksForGeeks-Branching and merging (master)
$
```

5. Các vấn đề thường gặp và cách xử lý

5.1 Viết sai message commit

- Để thay đổi nội dung commit hoặc thêm, sửa file vào commit: `git commit --amend`

```
File Edit View Search Terminal Help
GNU nano 2.9.3 /home/hydat/Study/Document/Do_an/backend/.git/COMMIT_EDITMSG Modified
commit 1 va commit 2
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sun Nov 28 16:32:01 2021 +0700
#
# On branch issue1
# Changes to be committed:
#   new file:   fix_issue_1.js
#   new file:   fix_issue_2.js
#
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text   M-J To Bracket  M-_ Previous
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell    ^_ Go To Line   M-E Redo       M-C Copy Text   M-W WhereIs Next M-^ Next
```

5. Các vấn đề thường gặp và cách xử lý

5.2 Đặt sai tên branch

- Để thay đổi tên branch hiện tại: `git branch -m <tên branch hiện tại> <tên branch sau khi đổi>`

```
File Edit View Search Terminal Help
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git status
On branch issue1
nothing to commit, working tree clean
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git branch -m issue1 issue2
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git status
On branch issue2
nothing to commit, working tree clean
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$
```


5. Các vấn đề thường gặp và cách xử lý

5.3 Revert commit

- Để revert một commit đầu tiên, có 2 trường hợp sau:
 - Khi commit đó chưa được push: `git reset HEAD~1`
 - Khi commit đó đã được push: `git revert <commit id>`

```
File Edit View Search Terminal Help
commit 540c4f4b06a42aec1f249b270ef6766f30783750 (HEAD -> issue2)
Author: hydat <pthd1998@gmail.com>
Date: Sun Nov 28 16:32:01 2021 +0700
```

commit 1 va commit 2

```
commit 803772ad0e6920af6b7d27b053893ee047565b9d (origin/backend, backend)
Author: hydat <pthd1998@gmail.com>
Date: Wed Jan 8 21:19:46 2020 +0700
```

add README file

```
File Edit View Search Terminal Help
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git reset HEAD~
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$
```

```
File Edit View Search Terminal Help
commit 803772ad0e6920af6b7d27b053893ee047565b9d (HEAD -> issue2, origin/backend, backend)
Author: hydat <pthd1998@gmail.com>
Date: Wed Jan 8 21:19:46 2020 +0700
```

add README file

5. Các vấn đề thường gặp và cách xử lý

5.4 Chuyển sang branch khác khi đang làm việc

Trường hợp đang làm việc trên branch A, nhưng sau đó muốn chuyển qua branch B để làm việc và sau khi kết thúc công việc ở branch B thì trở lại branch A.

- Lưu những thay đổi ở branch A: *git stash*
- Xem số thứ tự của lần lưu thay đổi ở trên: *git stash list*
- Sau khi trở lại branch A và tiếp tục làm việc: *git stash pop stash@{<số thứ tự>}*

```
File Edit View Search Terminal Help
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git status
On branch issue2
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   flx_issue_1.txt

no changes added to commit (use "git add" and/or "git commit -a")
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git stash
Saved working directory and index state WIP on issue2: b685276 add file 1
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git stash list
stash@{0}: WIP on issue2: b685276 add file 1
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git checkout backend
Switched to branch 'backend'
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git status
On branch backend
nothing to commit, working tree clean
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git checkout issue2
Switched to branch 'issue2'
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git status
On branch issue2
nothing to commit, working tree clean
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git stash pop stash@{0}
On branch issue2
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   flx_issue_1.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped stash@{0} (5d82efdf4b0747b7d94e245baf4774bf454bc9ab)
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git stash list
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$
```

5. Các vấn đề thường gặp và cách xử lý

5.4 Checkout sai branch

Ví dụ có 2 branch: A và B. Bạn muốn checkout một branch mới (branch C) từ branch A để code sau đó commit nhưng bạn lại checkout nhầm từ branch B.

- Chuyển sang branch A: *git checkout A*
- Tạo branch mới từ branch A: *git checkout -b D*
- Merge commit từ branch C qua branch D: *git cherry-pick <commit id>*

```
File Edit View Search Terminal Help
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git status
On branch issue2
nothing to commit, working tree clean
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git checkout -b issue3
Switched to a new branch 'issue3'
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git status
On branch issue3
nothing to commit, working tree clean
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git status
On branch issue3
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    fix_issue_3.txt

nothing added to commit but untracked files present (use "git add" to track)
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git add fix_issue_3.txt
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git commit -m"add file fix issue 3"
[issue3 1f26006] add file fix issue 3
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fix_issue_3.txt
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git checkout backend
Switched to branch 'backend'
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git checkout -b issue3_new
Switched to a new branch 'issue3_new'
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$ git cherry-pick 1f26006
[issue3_new 48b69f9] add file fix issue 3
Date: Sun Nov 28 16:46:53 2021 +0700
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fix_issue_3.txt
hydat@hydat-HP-EliteBook-840-G1:~/Study/Document/Do_an/backend$
```

5. Các vấn đề thường gặp và cách xử lý

5.5 Conflict code

Trường hợp khi push hoặc pull code bị conflict thì ta sử dụng tool 'meld' trên Linux để giải quyết.

A screenshot of a terminal window titled 'MINGW64:/f/Git_Branching_Demo'. The terminal shows a user named 'CodeTej' at 'LAPTOP-07TVC84J' in a 'MINGW64' environment. They are in the directory '/f/Git_Branching_Demo' and are on the 'navbar-new-feature' branch. They run the command '\$ git checkout master', which switches them to the 'master' branch. Then they run '\$ git merge navbar-new-feature', which attempts to merge the 'navbar-new-feature' branch into 'master'. This results in a conflict in 'index.html', with the message 'CONFLICT (content): Merge conflict in index.html' and 'Automatic merge failed; fix conflicts and then commit the result.' The terminal then shows they are in the 'master|MERGING' state, and the prompt '\$ |' is visible.

```
MINGW64:/f/Git_Branching_Demo

CodeTej@LAPTOP-07TVC84J MINGW64 /f/Git_Branching_Demo (navbar-new-feature)
$ git checkout master
Switched to branch 'master'

CodeTej@LAPTOP-07TVC84J MINGW64 /f/Git_Branching_Demo (master)
$ git merge navbar-new-feature
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

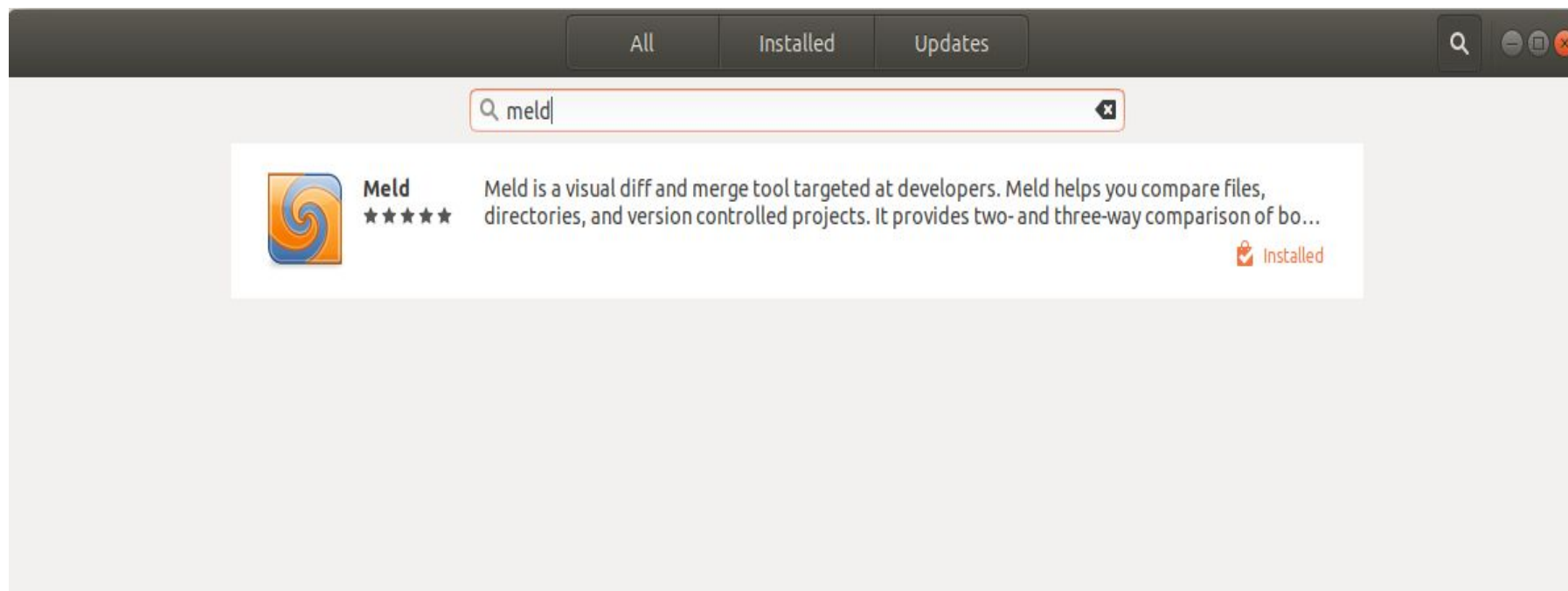
CodeTej@LAPTOP-07TVC84J MINGW64 /f/Git_Branching_Demo (master|MERGING)
$ |
```

5. Các vấn đề thường gặp và cách xử lý

5.5 Conflict code

Hướng dẫn cài tool 'meld' trên Linux.

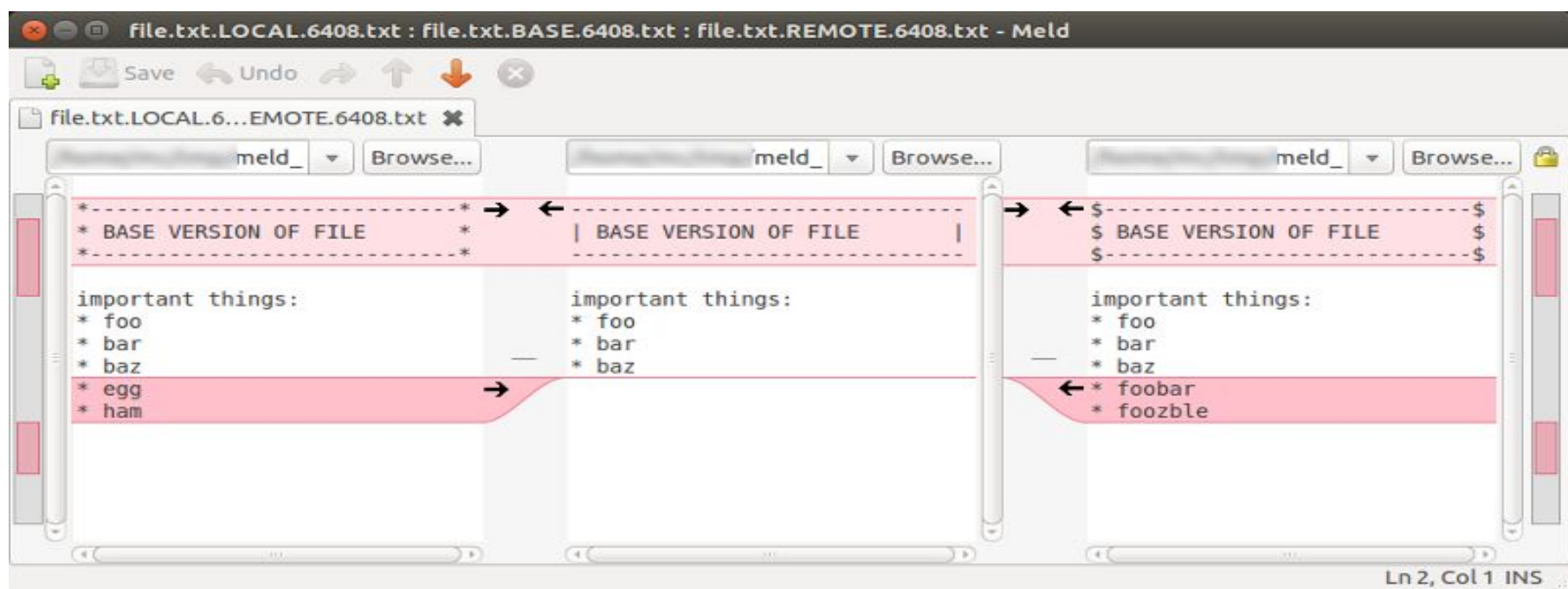
- Dùng command line: <https://www.unixmen.com/install-meld-on-ubuntu-and-mint-linux/>
- Dùng ubuntu software:



5. Các vấn đề thường gặp và cách xử lý

5.5 Conflict code

Dùng tool 'meld' để xử lý conflict. Cách dùng: `git mergetool --tool=meld`



KẾT THÚC