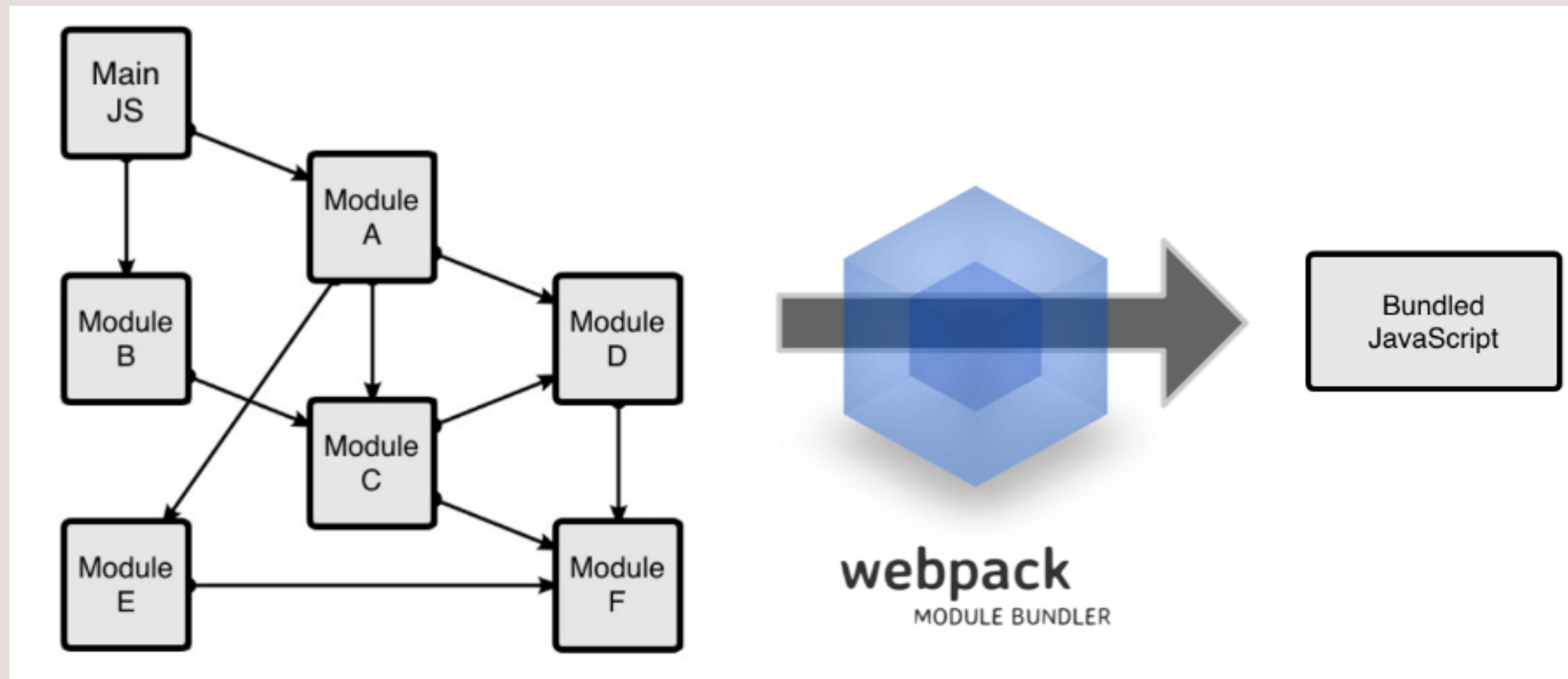
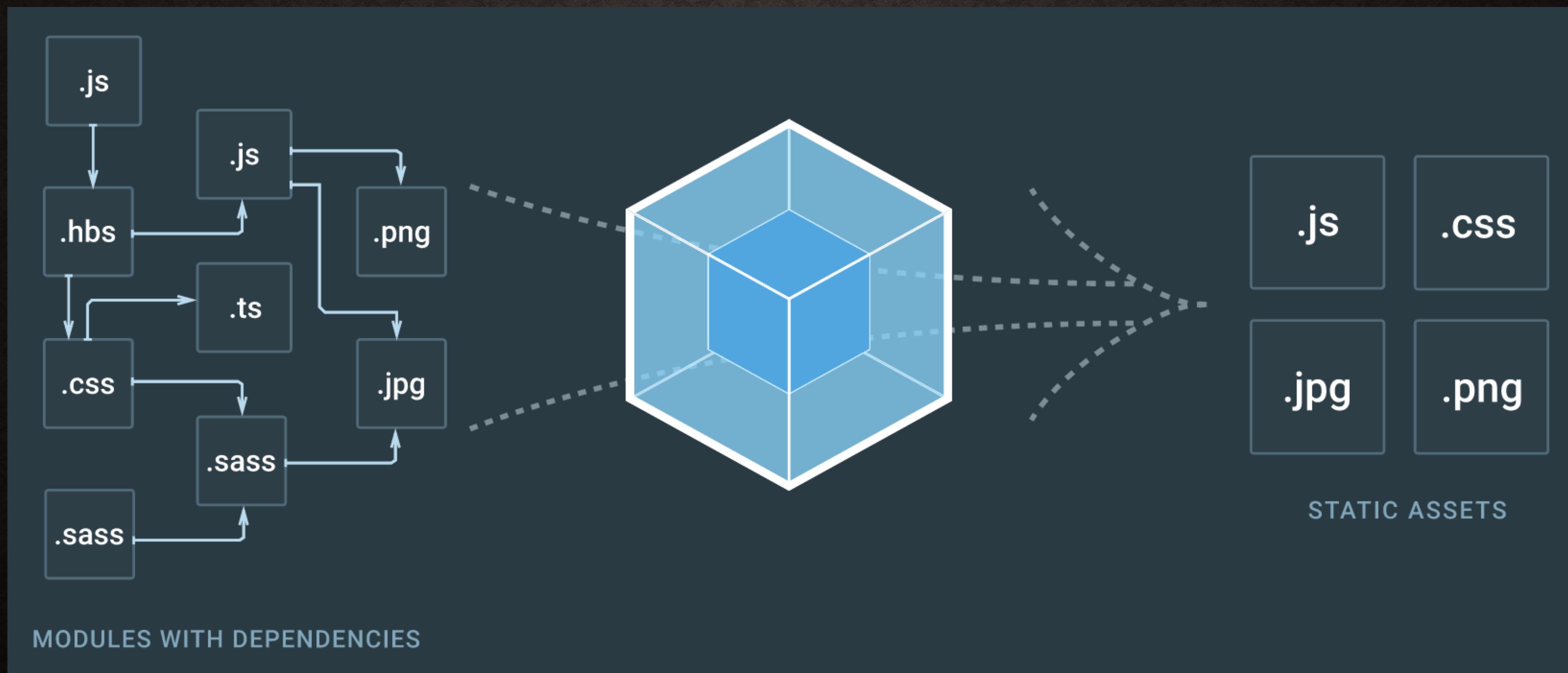


# Webpack



# Webpack



# Cài đặt webpack

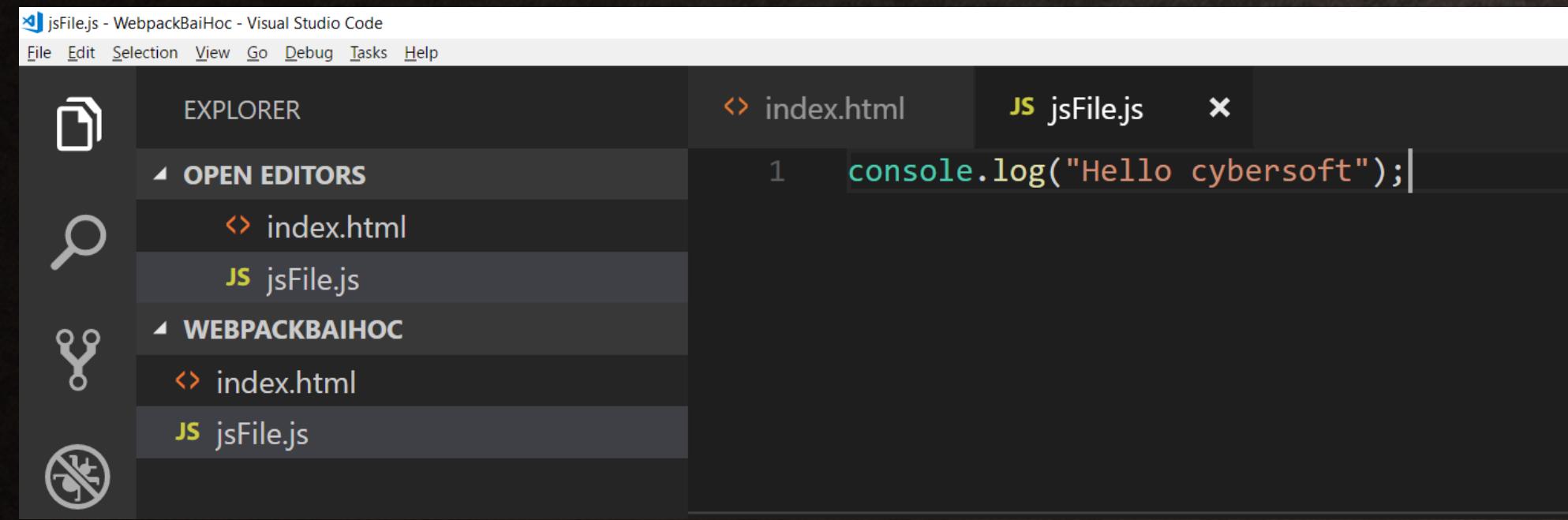
- + Cài nodejs trước (để chạy được cú pháp npm)
- + Mở project bằng visual code
- + Chọn Open in terminal
- + Gõ cú pháp : **npm install -g webpack** (để tiến hành cài đặt )
- + **npm install -g webpack-cli**

```
PS C:\Users\HP\Desktop\webpack> npm install -g webpack
C:\Users\HP\AppData\Roaming\npm\webpack -> C:\Users\HP\AppData\Roaming\npm\node_modules\webpack\bin\we
bpack.js
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3 (node_modules\webpack\node_modules\fsev
ents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.3: wanted {"os":"d
arwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

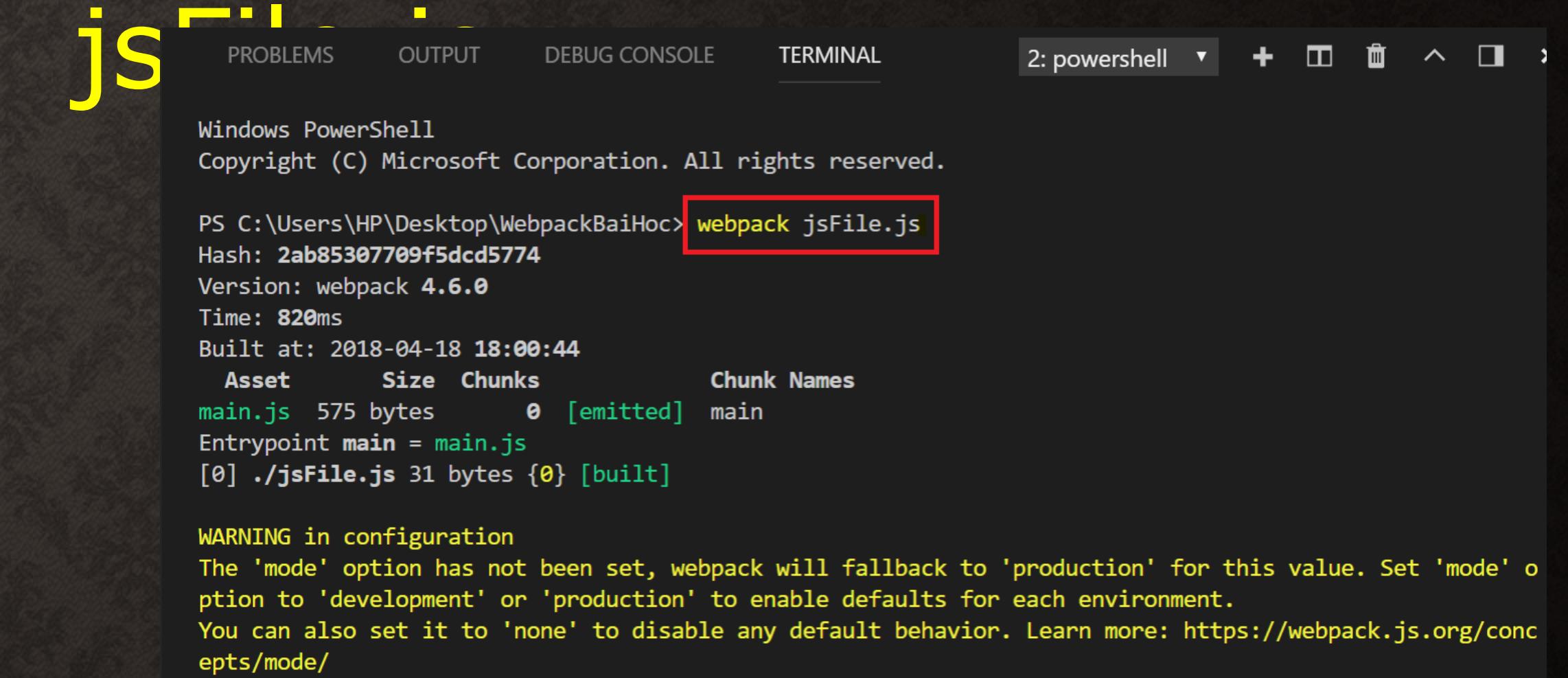
+ webpack@4.5.0
updated 4 packages in 9.616s
PS C:\Users\HP\Desktop\webpack> npm install -g webpack/cli
```

# Đóng gói javascript với webpack

Tạo file `jsFile.js` với nội dung + Gõ cú pháp : webpack



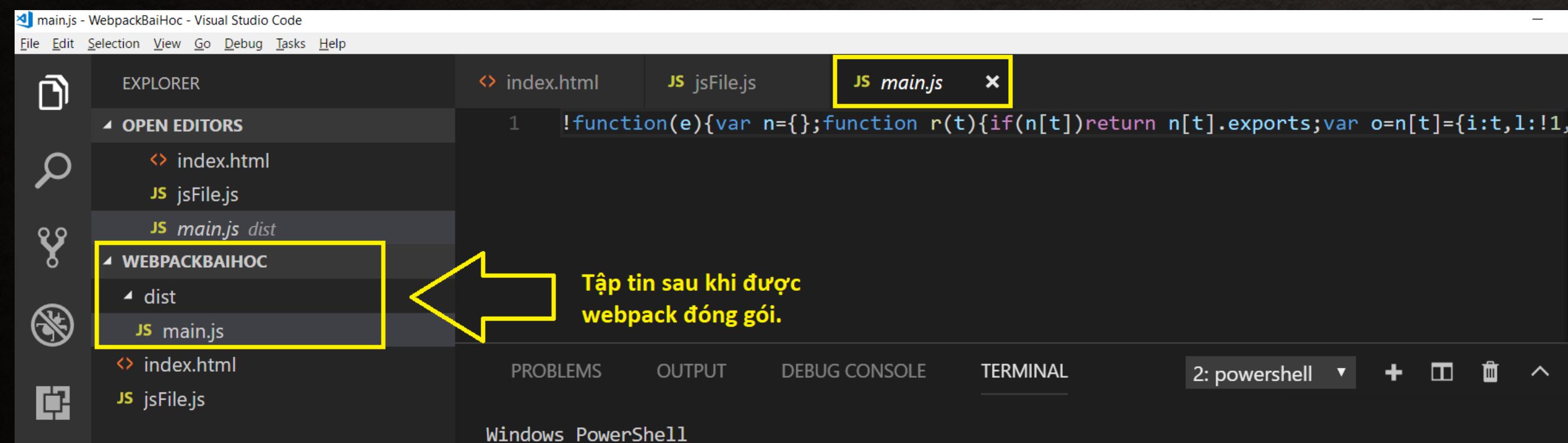
```
console.log("Hello cybersoft");
```



```
webpack jsFile.js
Hash: 2ab85307709f5dcd5774
Version: webpack 4.6.0
Time: 820ms
Built at: 2018-04-18 18:00:44
Asset      Size  Chunks      Chunk Names
main.js    575 bytes    0  [emitted]  main
Entrypoint main = main.js
[0] ./jsFile.js 31 bytes {0} [built]

WARNING in configuration
The 'mode' option has not been set, webpack will fallback to 'production' for this value. Set 'mode' option to 'development' or 'production' to enable defaults for each environment.
You can also set it to 'none' to disable any default behavior. Learn more: https://webpack.js.org/concepts/mode/
```

Cú pháp trên để đóng gói `jsFile.js` => `main.js` inline 1 dòng



```
!function(e){var n={};function r(t){if(n[t])return n[t].exports;var o=n[t]={i:t,l:!1,exports:{}};e[t](function(t){Object.defineProperty(o,'exports',{value:t})}),n[t]=o};return r}();
```

Tập tin sau khi được webpack đóng gói.

# Đóng gói nhiều file js với webpack

Ví dụ để đóng gói **2 prototype + main.js**:

```
JS DanhSachSinhVien.js ×  
1 export function DanhSachSinhVien()  
2 {  
3     this.DSSV = [];  
4     this.ThemSinhVien = function (sinhvienthem)  
5     {  
6         this.DSSV.push(sinhvienthem);  
7     }  
8 }
```

```
JS DanhSachSinhVien.js JS SinhVien.js ×  
1 export function SinhVien(hoten_, email, cmnd, sdt)  
2 {  
3     this.HoTen = hoten_;  
4     this.Email = email;  
5     this.CMND = cmnd;  
6     this.SoDT = sdt;  
7 }
```

Trong **main.js** ta cần sử dụng 2 prototype đó.

```
JS main.js ×  
1 import {DanhSachSinhVien} from './DanhSachSinhVien.js';  
2 import {SinhVien} from './SinhVien.js';  
3  
4 //Khởi tạo 1 danh sách sinh viên  
5 var danhSachSinhVien = new DanhSachSinhVien();  
6 SinhVien.prototype.MaSV = '';  
7  
8
```

Để đóng gói được ta cần **export** các prototype ra. Để file js nào cần ta có thể **import** vào.

# Đóng gói nhiều file js với webpack

Lưu ý: Đối với các function ta cũng có thể export function ra ngoài.

```
export function click(){
|   console.log('day la file main3.js')
}
|
```

```
|  
import './main3';  
import { click } from './main3';
```

# Đong gói nhiều file JS với webpack

Câu hỏi đặt ra vậy lõi 1 file có nhiều prototype thì sử dụng như thế nào ? Để giải quyết điều đó ta có khái niệm về module Export giúp ta có thể export nhiều function trong cùng 1 file

## Ví dụ

```
index.html      JS main.js      JS Test.js x
1  function HocSinh(name)
2  {
3      this.name = name;
4  }
5  function TruongHoc(name,address)
6  {
7      this.name = name;
8      this.address = address;
9  }
10 //Export nhiều function cùng lúc
11 module.exports = {HocSinh:HocSinh,TruongHoc:TruongHoc};
```

Key ↓ Function name ↓

```
index.html x      JS main.js x      JS Test.js
185 //Để sử dụng ta cần import các prototype từ file test.js vào main.js
186 import {HocSinh,TruongHoc} from './Test.js';
187
188 var hs = new HocSinh('abc');
189 console.log(hs);
190
191 var th = new TruongHoc('Cybersoft','452 Sư Vạn Hạnh');
192 console.log(th);
```

Để sử dụng trong file main ta import Key từ đối tượng module.exports

Ta có 1 file Test.js Chứa 2 prototype HocSinh và TruongHoc

# Đóng gói css với module style-loader

- ❑ Cài đặt module style loader

```
npm install style-loader css-loader --save-dev
```

# Tạo file webpack.config.js

```
const path = require('path');
module.exports = {
entry: './js/main.js',
devtool: 'source-map',
module: {
rules: [
//SASS LOADER ĐỂ ĐÓNG GÓI CÁC FILE SASS
{
test: /\.scss$/,
use: [
{loader: "style-loader"},
{
loader: "css-loader",
options:{minimize:true,
},
},
{loader: "sass-loader"},
]
},
],
},
};

//CSS LOADER ĐỂ ĐÓNG GÓI CÁC FILE CSS
{
test:/\.css$/,
use:[{loader: "style-loader"},
{
loader: "css-loader"
}
],
```

=> Sau khi cấu hình xong gõ cú pháp : **webpack** đ

# Cú pháp đóng gói CSS

```
//CSS LOADER ĐỂ ĐÓNG GÓI CÁC FILE CSS
{
  test: /\.css$/,
  use: [ {loader: "style-loader"} ,
  {
    loader: "css-loader",
    options: {
      minimize: true,
    }
  } ] ,
},
```

# Đóng gói sass với module style-loader

- Cài đặt module node-sass

Cú pháp:

```
npm install node-sass sass-loader --save-dev
```

```
{  
  test: /\.scss$/,
  use: [
    {loader: "style-loader",
    {
      loader: "css-loader",
      options: {
        minimize:true,
      }
    },
    {loader: "sass-loader"},
  ],
},
```

=> Sau khi cấu hình xong gõ cú pháp : webpack để build

# Đóng gói hình ảnh với module

- Cài đặt module node-sass

Cú pháp:

```
npm install file-loader url-loader --save -dev
```

```
{  
  test: /\.(\jpg|\png)$/,  
  use: {  
    loader: "url-loader",  
    options: {  
      name: ".../[path][name].[ext]",  
      limit:1000,  
    },  
  },  
},  
},
```

=> Sau khi cấu hình xong gõ cú pháp : **webpack** để **build**

# Quản lý mã nguồn



# Version control system?

- Hệ thống giúp lưu trữ source code và có thể theo dõi sự thay đổi của code.
- Hỗ trợ nhiều người dùng làm việc trên cùng 1 source
- Kiểm tra những thay đổi và biết được ai là người thay đổi
- Revert các thay đổi, đưa code trở về các phiên bản cũ, không lo bị mất code

# Git là gì ?

- ❑ Git là một phần mềm, một trong những VCS , vốn được phát triển nhằm quản lý mã nguồn của Linux.
- ❑ Mỗi người sẽ tự làm việc trên máy tính của mình, sau khi thay đổi sẽ tiến hành update lên source code chung thông qua một số quy trình
- ❑ Là một trong những VCS tốt nhất hiện tại
- ❑ Các khái niệm chính trong git :
  - ❖ Repository: kho lưu trữ source code và tất cả các thay đổi
    - Local repo: kho chứa trên máy tính chúng ta, mỗi máy sẽ là một repo
    - Remote repo : kho dữ liệu chứa trên server, có thể truy cập từ bất kì đâu  
=> dữ liệu ở local có thể được push lên remote để lưu trữ chung  
dữ liệu ở remote có thể được clone về local, mang theo tất cả dữ liệu và thiết lập có sẵn
  - ❖ Commit: mỗi lần thay đổi source code, tiến hành commit để tạo ra 1 version lưu trữ lại sự thay đổi, khi có lỗi thì có rollback về version trước đó đc

# Cài đặt git – git-bash

Linux:: <http://git-scm.com/download/>

mac

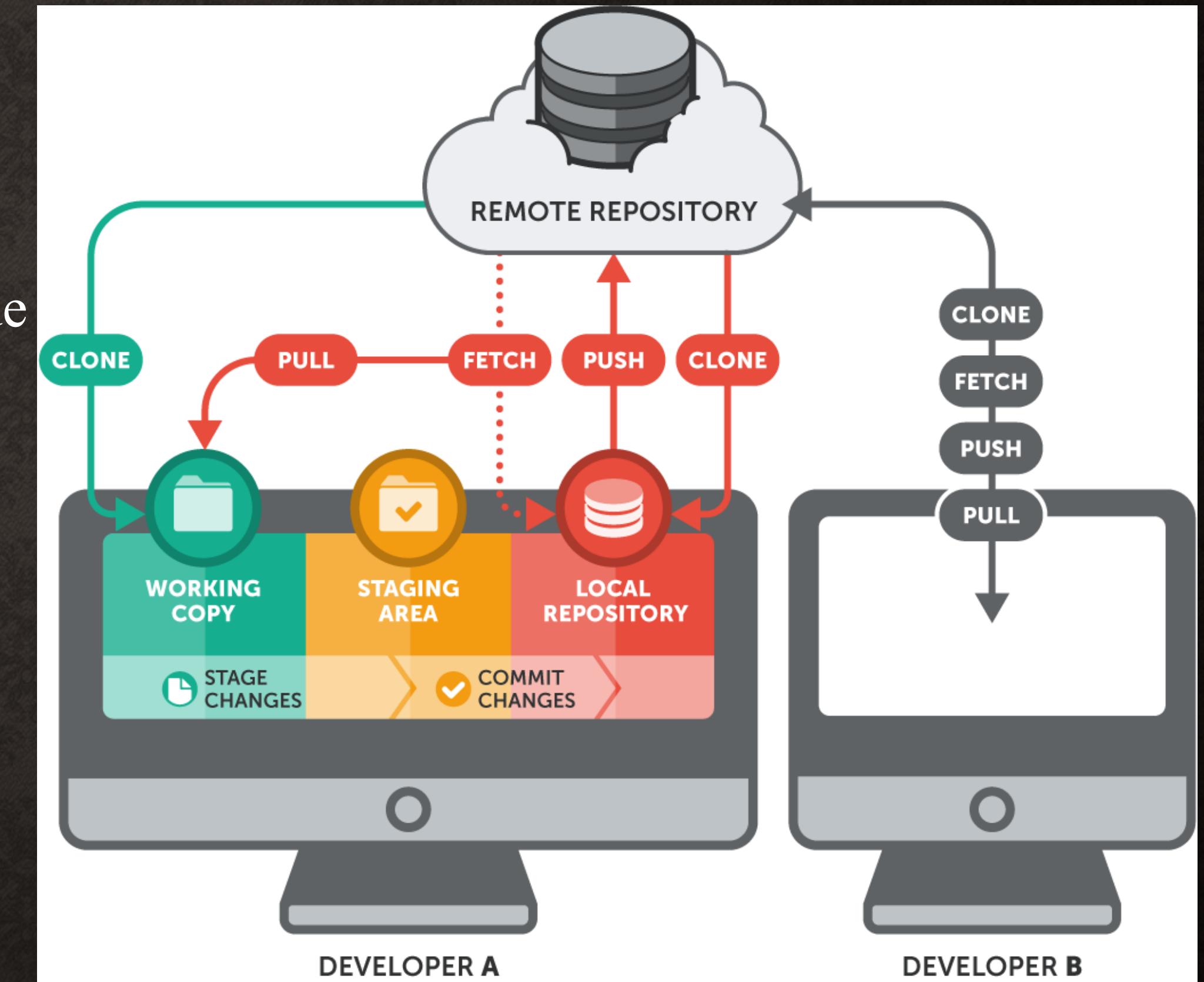
Mac: <http://git-scm.com/download/mac>

Windows: <http://git-scm.com/download/>

win

# Git hub là gì ?

- Server lưu trữ các remote repo
- Cho phép nhiều người có thể làm việc trên cùng 1 source code
- Sơ đồ hoạt động giữa git và git hub



- ❑ Thực hành với git:
  - ❑ Tạo dự án trên git

Bước 1: Đăng ký 1 tài khoản trên github <https://github.com/login>

Bước 2: Đặt tên tài khoản Repository name

Create a new repository  
A repository contains all the files for your project, including the revision history.

Owner: kaitokido14120 / Repository name: DuAnGit

Great repository names are short and memorable. Need inspiration? How about [special-octo-chainsaw](#).

Description (optional): Dự án git demo

**Public** ↖  
Anyone can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None | ⓘ

**Create repository**

Chọn mục public vì private có thu phí

=> Create repository

- ❑ Thực hành với git:
  - ❑ Tạo dự án trên git

Bước 1: Đăng ký 1 tài khoản trên github <https://github.com/login>

Bước 2: Đặt tên tài khoản Repository name

Create a new repository  
A repository contains all the files for your project, including the revision history.

Owner: kaitokido14120 / Repository name: DuAnGit

Great repository names are short and memorable. Need inspiration? How about [special-octo-chainsaw](#).

Description (optional): Dự án git demo

**Public** *(Red arrow points here)*  
Anyone can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None | ⓘ

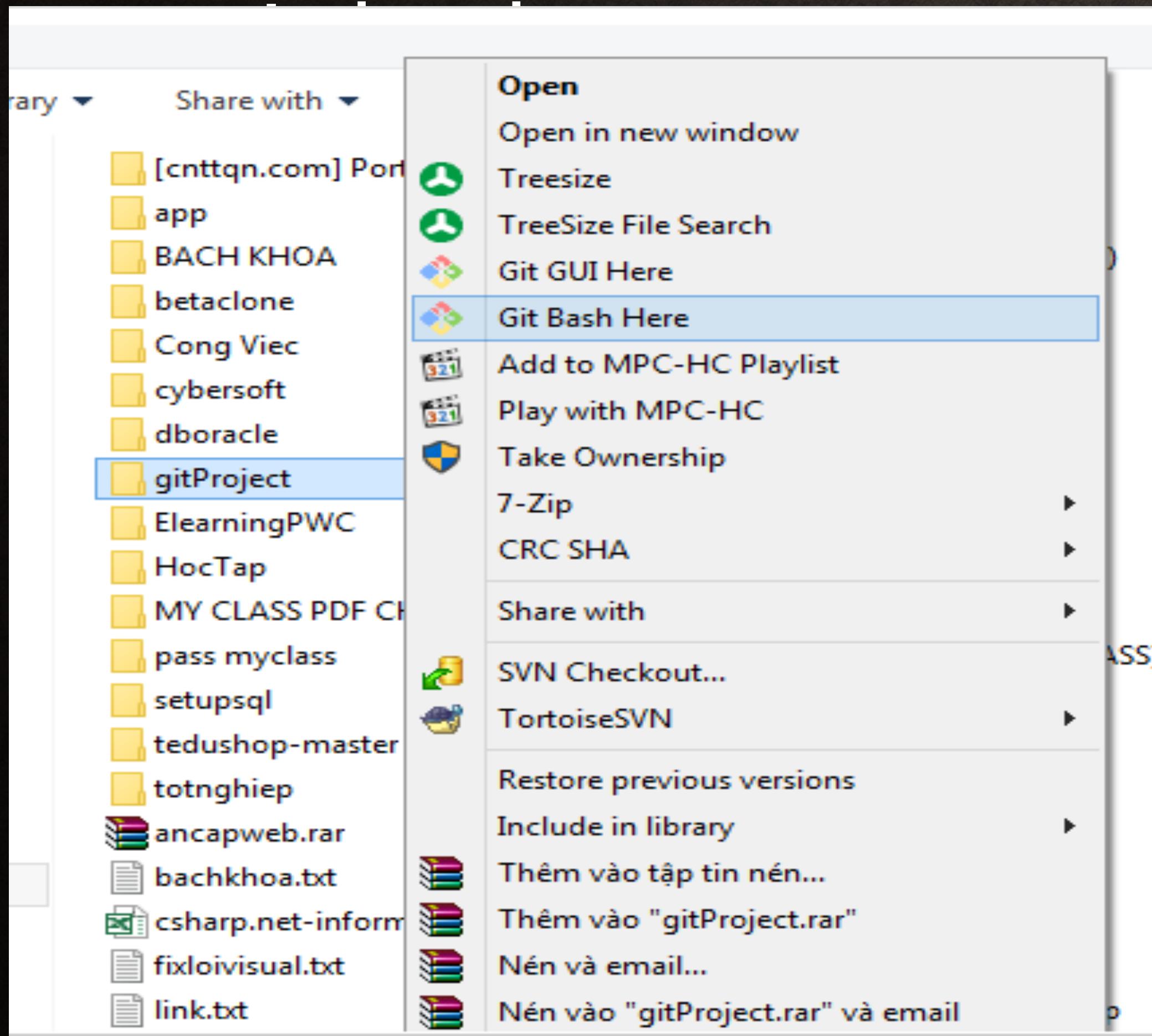
**Create repository**

Chọn mục public vì private có thu phí

=> Create repository

- ❑ Thực hành với git:
  - ❑ Lấy dự án từ git về

Bước 1: Chọn folder chứa project, chuot phải chọn



A screenshot of a terminal window titled 'MINGW32:/e/gitProject'. The window shows a command prompt with the text 'Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject \$ |'. The terminal is currently empty, awaiting input.

- ❑ Đối với người khởi tạo dự án, thực hiện các lệnh sau

Dùng lệnh **git init** để khởi tạo

Cấu hình tài khoản tại local : **git config --global user.email "dangtrunhieu"**

Dùng lệnh **git remote add origin** “Đường dẫn trên git” để trỏ tới server

// Dùng lệnh lần đầu để trỏ đến host chứa project

Dùng lệnh **git add [tên file]** để add file cần upload lên git

Dùng lệnh **git commit -m “ghi chú”** để ghi chú

Dùng lệnh **git push -u origin master** để đẩy dữ liệu upload lên

# ❑ Thêm user vào dự án trên github

The screenshot shows a GitHub repository page for 'kaitokido14120 / DuAnGit'. The top navigation bar includes links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the repository name, there are buttons for 'Watch' (0), 'Star' (0), and 'Fork' (0). The main navigation tabs are 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings' (which is highlighted with a red arrow), and 'Insights'. On the left, a sidebar menu lists 'Options', 'Collaborators' (which is also highlighted with a red arrow), 'Branches', 'Webhooks', 'Integrations & services', and 'Deploy keys'. The main content area is titled 'Collaborators' and contains a message 'Awaiting myclass0102's response' with 'Copy invite link' and 'Cancel invite' buttons. It also features a search bar for finding users by username, full name, or email address, with a note explaining that email addresses must be public. At the bottom, there are links for 'Contact GitHub', 'API', 'Training', 'Shop', 'Blog', and 'About'.

This repository

Pull requests Issues Marketplace Explore

kaitokido14120 / DuAnGit

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Options Collaborators Branches Webhooks Integrations & services Deploy keys

Collaborators

Awaiting myclass0102's response

Push access to the repository

Copy invite link Cancel invite

Search by username, full name or email address

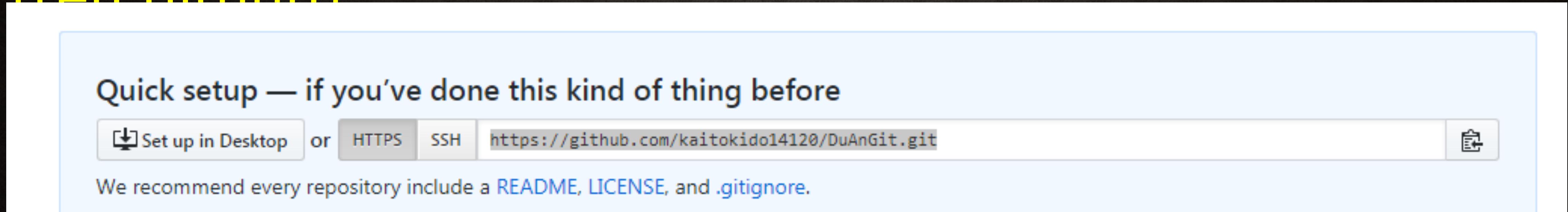
You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Add collaborator

© 2017 GitHub, Inc. Terms Privacy Security Status Help

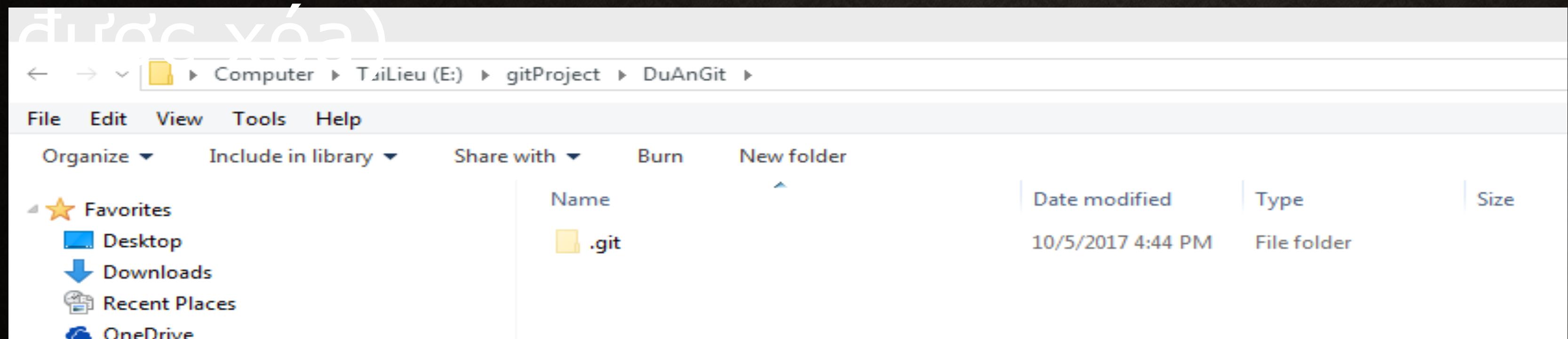
Contact GitHub API Training Shop Blog About

# Đối với người tham gia dự án : gõ lệnh git clone '[url trên github]'



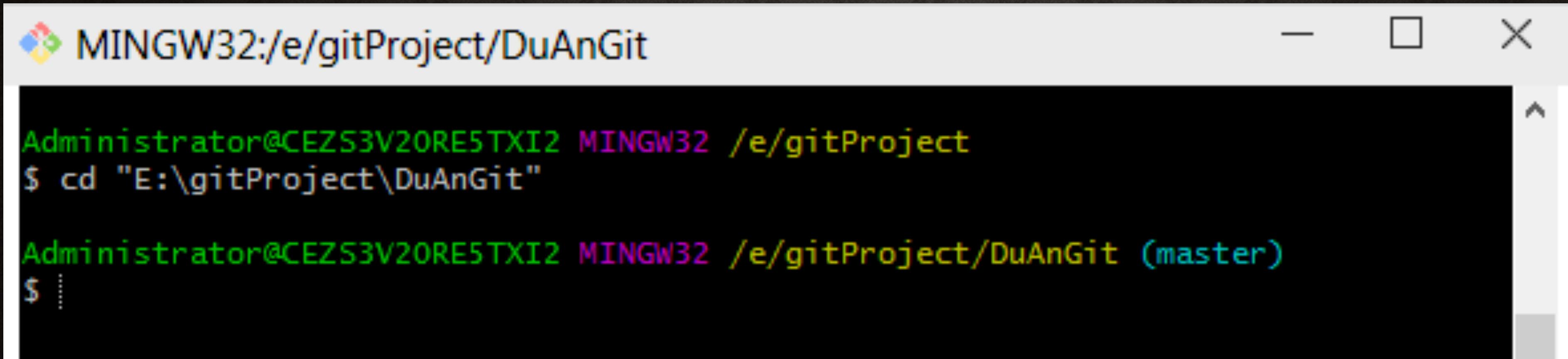
A screenshot of a terminal window titled 'MINGW32:/e/gitProject'. The command \$ git clone "https://github.com/kaitokido14120/DuAnGit.git" is being typed in.

## Sau khi chạy xong ta có được folder quản lý git (Không được xóa)



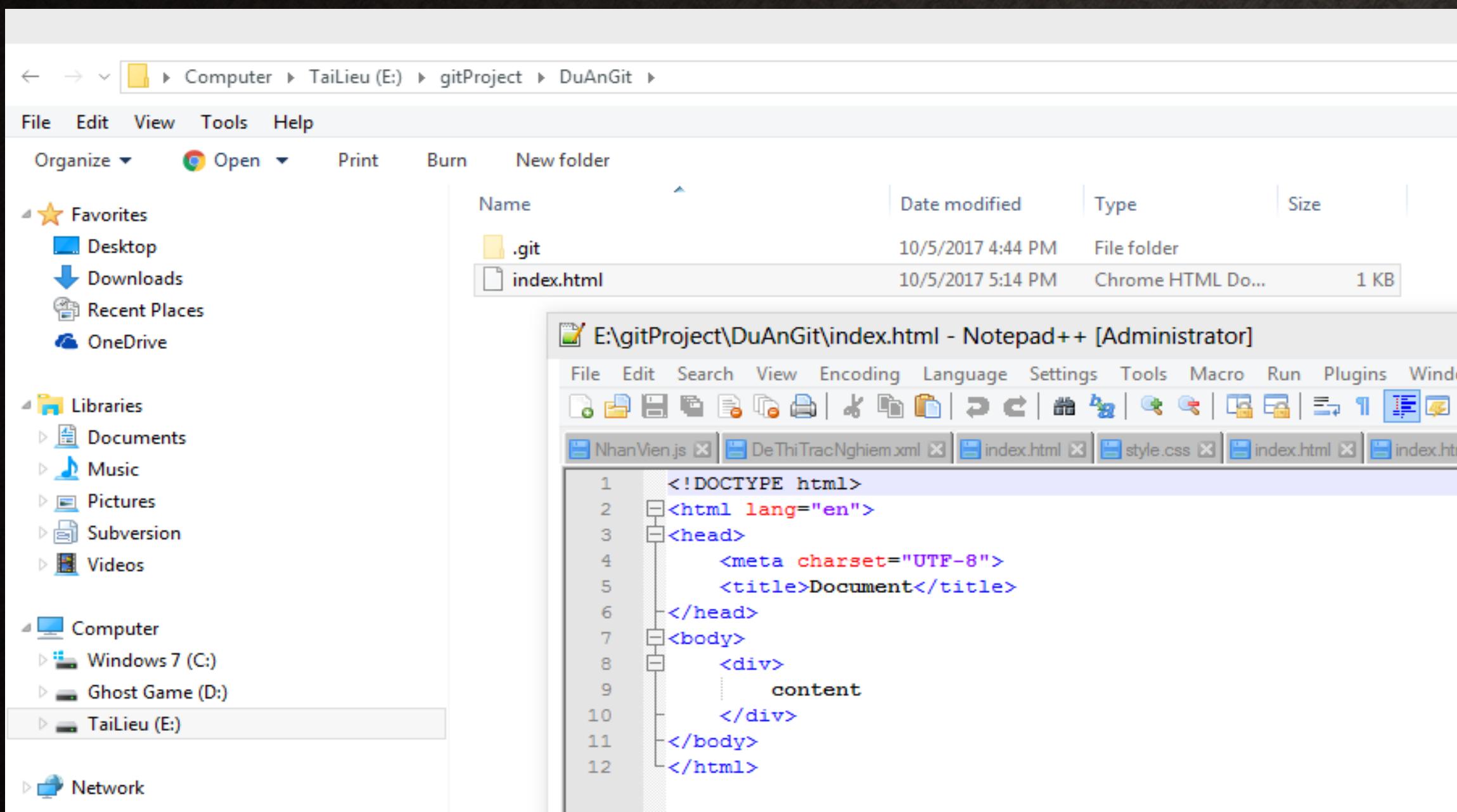
# □ Phát triển project -> upload lên git

Bước 1: Trỏ đến folder cần upload các file code



```
MINGW32:/e/gitProject/DuAnGit
Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject
$ cd "E:\\gitProject\\DuAnGit"

Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ ..
```



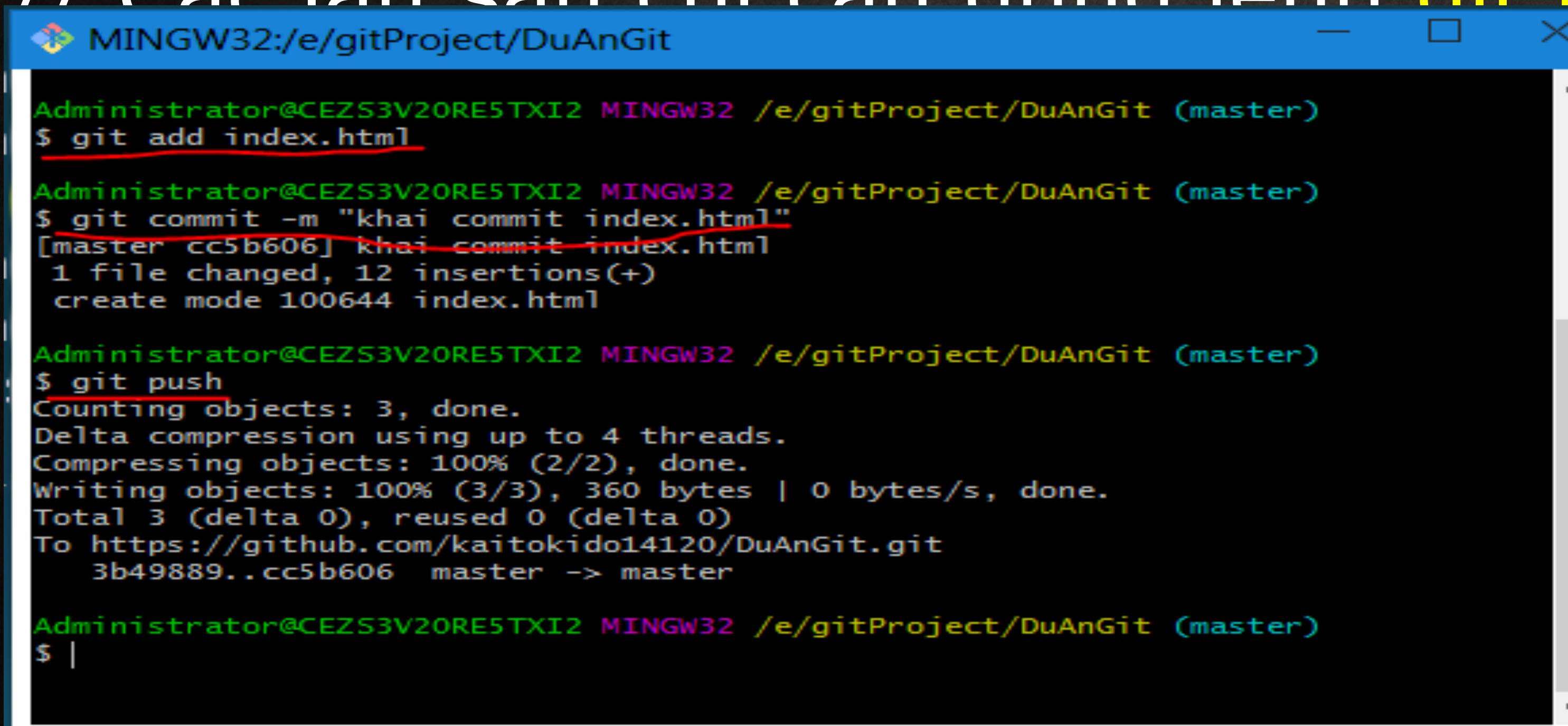
Dùng lệnh cd “[đường dẫn]”

Ở đây là thư mục gốc

## Bước 2:

Dùng lệnh `git add [tên file]` để add file cần upload lên git  
Dùng lệnh `git commit -m "ghi chú"` để ghi chú  
Dùng lệnh `git push -u origin master` để đẩy dữ liệu upload lên

// Các lần sau chỉ cần dùng lệnh `git push`



```
MINGW32:/e/gitProject/DuAnGit
Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ git add index.html

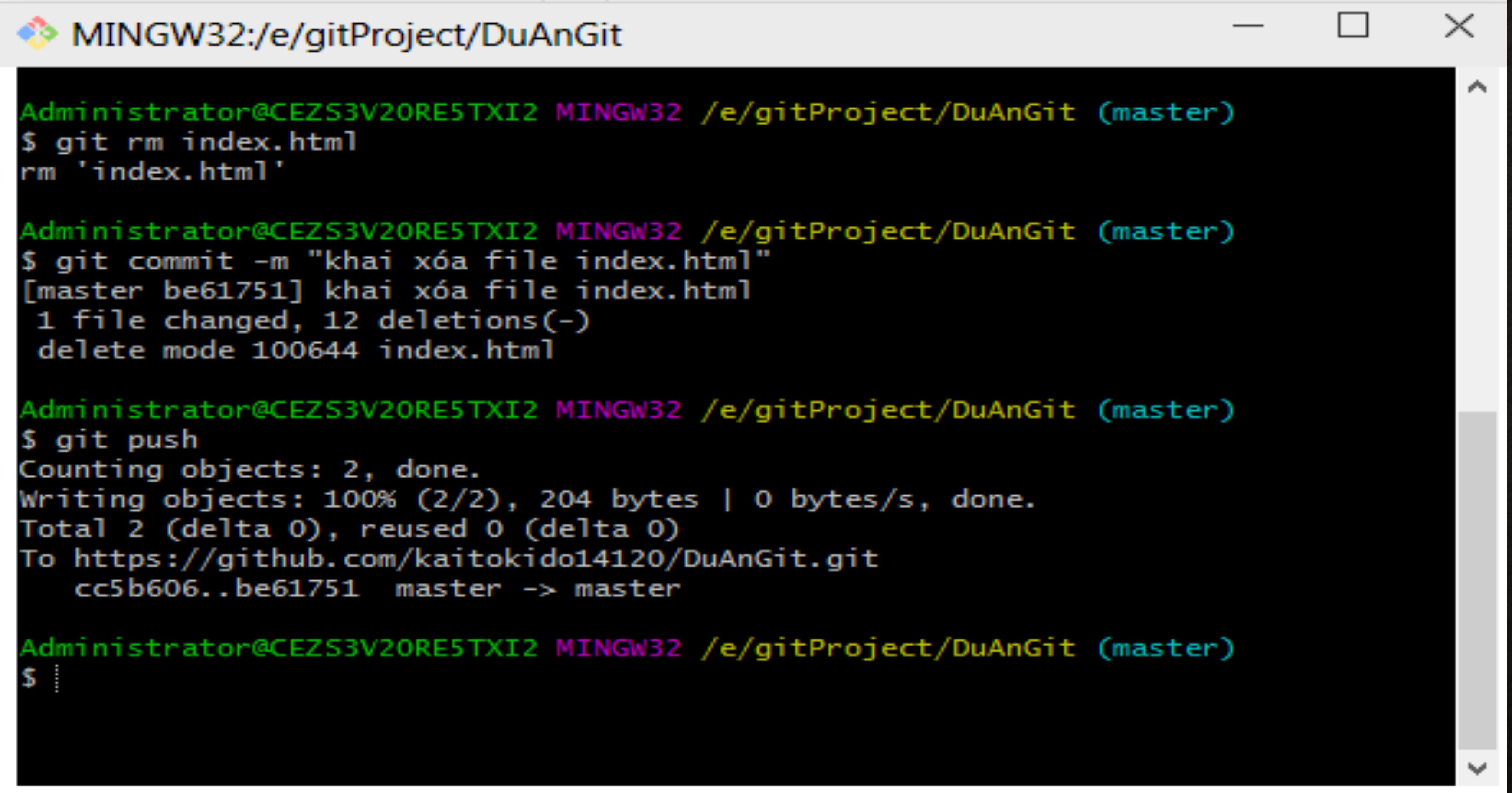
Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ git commit -m "khai commit index.html"
[master cc5b606] khai commit index.html
 1 file changed, 12 insertions(+)
 create mode 100644 index.html

Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 360 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/kaitokido14120/DuAnGit.git
 3b49889..cc5b606  master -> master

Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ |
```

## ☐ Remove file trên git

Dùng lệnh git rm index.html



The screenshot shows a terminal window titled "MINGW32:/e/gitProject/DuAnGit". The terminal output is as follows:

```
Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ git rm index.html
rm 'index.html'

Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ git commit -m "khai xóa file index.html"
[master be61751] khai xóa file index.html
 1 file changed, 12 deletions(-)
 delete mode 100644 index.html

Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ git push
Counting objects: 2, done.
Writing objects: 100% (2/2), 204 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To https://github.com/kaitokido14120/DuAnGit.git
 cc5b606..be61751 master -> master

Administrator@CEZS3V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ ..
```

## ❑ Commit nhiều file

Dùng lệnh git add -A (-A: là commit all file trong thư mục đang

```
MINGW32:/e/gitProject/DuAnGit
Administrator@CEZ53V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ git add -A

Administrator@CEZ53V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ git commit -m "commit html css js"
[master 41a7172] commit html css js
 5 files changed, 12 insertions(+)
 create mode 100644 css/style01.css
 create mode 100644 css/style02.css
 create mode 100644 html/index.html
 create mode 100644 js/js01.js
 create mode 100644 js/js02.js

Administrator@CEZ53V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ git push
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 568 bytes | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/kaitokido14120/DuAnGit.git
  be61751..41a7172  master -> master

Administrator@CEZ53V20RE5TXI2 MINGW32 /e/gitProject/DuAnGit (master)
$ |
```

## ❑ Update (cập nhật dữ liệu từ git xuống project trên máy )

### **git clone**

Lệnh này sẽ sao chép toàn bộ dữ liệu trên repository và sao chép luôn các thiết lập về repository, tức là nó sẽ tự động tạo một master branch trên máy tính của bạn. Lệnh này chỉ nên sử dụng khi bạn cần tạo mới một Git mới trên máy tính với toàn bộ dữ liệu và thiết lập của một remote repository.

### **git pull**

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

# ❑ Branch là gì ?

- Mặc định khi làm việc với git, ta sẽ làm việc trên nhánh Master
- Khi có quá nhiều người cùng làm việc trên một source code, dễ dẫn tới khó khăn
- Ta phải tiến hành chia nhánh, mỗi nhánh thực hiện một chức năng khác nhau, sau đó gộp lại vào master, nhằm mục đích giữ nhánh chính (master) được toàn vẹn
- Một số cú pháp:
  - Cú pháp khởi tạo nhánh: **git checkout –b tenBranch**

## ❑ Link tham khảo thêm

<https://viblo.asia/p/tap-hop-nhung-cau-lenh-git-huu-dung-dWrvwWr2vw38>

<http://rogerdudler.github.io/git-guide/index.vi.html>

<https://www.codehub.vn/Khac-nhau-giua-Pull-va-Pull-Request-trong-Git>

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

```
echo "# WebsiteHoChieu" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/kaitokido14120/WebsiteHoChieu.git
git push -u origin master
```

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

```
git remote add origin https://github.com/kaitokido14120/WebsiteHoChieu.git
git push -u origin master
```