

# Getting started with Playwright

## Lesson 1



# Agenda

1. Hiii!
2. Tinh thần học
3. Các công cụ hỗ trợ học tập
4. Cài đặt: VSCode, kết nối GitHub
5. Tìm hiểu: Playwright và so sánh với công cụ khác
6. Cài đặt, chạy test Playwright đầu tiên

# Hi

Xin chào ^^



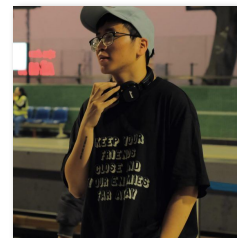
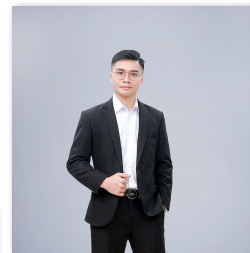
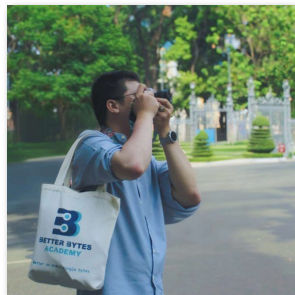
# Playwright Viet Nam Team

- **Mentors:**

- Minh Phong
- Quỳnh Nga
- Phạm Uyên
- Minh Thái
- Anh Lợi
- Quân Lê

- **Trợ giảng**

- Hường
- Ngân
- Ánh
- Thuỳ



# Playwright Viet Nam Team

- **Kinh nghiệm giảng dạy:**
  - ~ 800 học viên qua tất cả các khóa
- **Full-stack Playwright TypeScript:** 18 khoá + lớp riêng cho các đơn vị, công ty ~ 600 học viên
- **Playwright Master Class:** 80 học viên
- **API Testing:** ~150 học viên

# Playwright Viet Nam Team

- Lộ trình [rõ ràng, chi tiết](#)
- Các buổi học có recording đầy đủ

## [Lesson 6: Git and Javascript advance](#)



Slide lesson 06



Bài tập lesson 06



Recording - part 2



Recording - part 1

## Lesson 5: DOM terminology, Playwright



Correct exercise



Key Takeaway



Recording lesson 5



Bài tập lesson 5

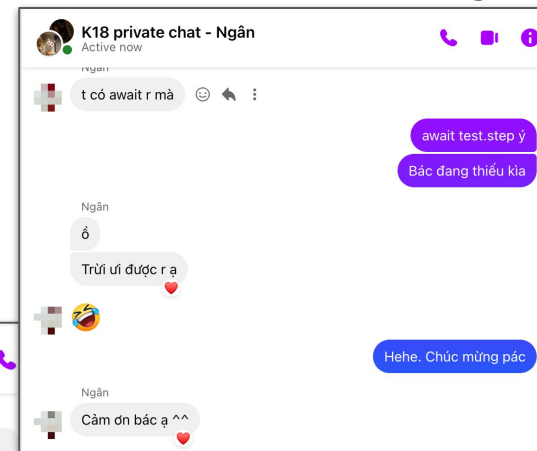
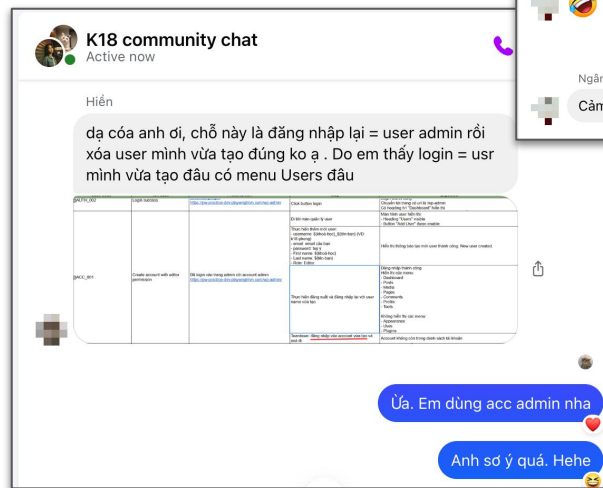
# Playwright Viet Nam Team

- Lộ trình [rõ ràng, chi tiết](#)
- Các buổi học có recording đầy đủ
- [Có chứng chỉ](#)
  - *Lưu ý*: hoàn thiện đầy đủ bài tập để nhận chứng chỉ



# Playwright Viet Nam Team

- Lộ trình [rõ ràng, chi tiết](#)
- Các buổi học có recording đầy đủ
- [Có chứng chỉ](#)
- Hỗ trợ liên tục thông qua [chat chung, chat riêng](#)



# Chào mừng bạn



# HELLO and welcome



Khởi động



Chậm mà chắc



Vượt chướng ngại vật



Tăng tốc

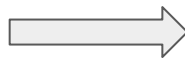


Về đích

Chương trình học chia thành  
**5 giai đoạn** với từng mục  
tiêu rõ ràng



**Khởi động**



**Cài đặt máy (buổi 1)**



**Chậm mà chắc**



**Vượt chướng ngại vật**



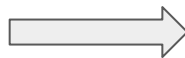
**Tăng tốc**



**Về đích**



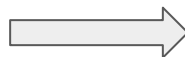
**Khởi động**



**Cài đặt máy**



**Chậm mà chắc**



**Kiến thức nền tảng  
(buổi 1 -> 4)**



**Vượt chướng ngại vật**



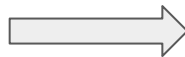
**Tăng tốc**



**Về đích**



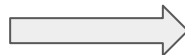
**Khởi động**



**Cài đặt máy**



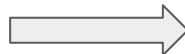
**Chậm mà chắc**



**Kiến thức nền tảng**



**Vượt chướng ngại vật**



**Kiến thức automation  
(buổi 5 -> 9)**



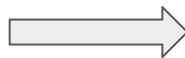
**Tăng tốc**



**Về đích**



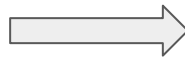
**Khởi động**



**Cài đặt máy**



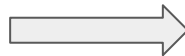
**Chậm mà chắc**



**Kiến thức nền tảng**



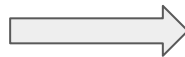
**Vượt chướng ngại vật**



**Kiến thức automation**



**Tăng tốc**



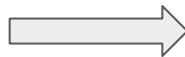
**Tối ưu code  
(buổi 10 -> 13)**



**Về đích**



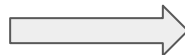
**Khởi động**



**Cài đặt máy**



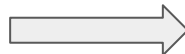
**Chậm mà chắc**



**Kiến thức nền tảng**



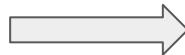
**Vượt chướng ngại vật**



**Kiến thức automation**



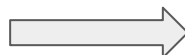
**Tăng tốc**



**Tối ưu code**



**Về đích**



**Setup project, CI CD  
(buổi 14, 15)**

# Tinh thần học

Học thế nào cho hiệu quả?



# Tinh thần học



**Học đúng cách** giúp nâng hiệu quả của khoá học **lên 65%**

Tại Better Bytes Academy, học đúng cách thông qua 3 tiêu chí

- Học **chủ động**
- Học **chăm chỉ**
- Học **thông minh**

# Tinh thần học



Học **chủ động**:

chủ động "ghi nhớ" kiến thức

- **Gặp vấn đề**: chủ động research thông qua Google/ AI tools trước.



# Tinh thần học



Học **chủ động**:

chủ động "ghi nhớ" kiến thức

- **Gặp vấn đề**: chủ động research thông qua Google/ AI tools trước.
- **Vẫn chưa tự giải quyết được**: tìm tới mentors trong chat chung/private chat



# Tinh thần học



Học **chủ động**:

chủ động "ghi nhớ" kiến thức

- **Gặp vấn đề**: chủ động research thông qua Google/ AI tools trước.
- **Vẫn chưa tự giải quyết được**: tìm tới mentors trong chat chung/private chat
- **Tìm hiểu thêm** các kiến thức mở rộng



# Tinh thần học



Học **chăm chỉ**:

Làm đầy đủ bài tập trước buổi học



# Tinh thần học



Học **chăm chỉ**:

Làm đầy đủ bài tập trước buổi học

- Thực hành giúp:
  - **Củng cố** lý thuyết
  - Phát triển **kỹ năng tư duy**



# Tinh thần học



Học **chăm chỉ**:

Làm đầy đủ bài tập trước buổi học

- Thực hành giúp:
  - **Củng cố** lý thuyết
  - Phát triển **kỹ năng tư duy**
- **Thực hành** => gặp lỗi => kinh nghiệm



# Tinh thần học



Học **thông minh**:

Giúp việc học hiệu quả hơn



# Tinh thần học



Học **thông minh**:

Giúp việc học hiệu quả hơn

- Luôn hệ thống lại các kiến thức đã học trước khi làm bài tập



# Tinh thần học



Học **thông minh**:

Giúp việc học hiệu quả hơn

- Luôn hệ thống lại các kiến thức đã học trước khi làm bài tập
- Tự hệ thống giúp nhớ bài, hiểu bài hơn



# Tinh thần học



Học **thông minh**:

Giúp việc học hiệu quả hơn

- **Không sử dụng AI để làm bài hộ**. Học bản chất cần dành thời gian suy nghĩ.



# Ví dụ file key takeaways

## 🔗 Playwright

- Là một framework tiền thân là Puppeteer, được Microsoft tài trợ và phát triển.
- <https://playwright.dev/>
- Ra đời: 31/01/2020

## Ưu điểm

- Cross browser
- Cross platform
- Auto-wait
- Web-first assertion
- Tracing
- Code-gen
- ...

# Công cụ học

Các công cụ sử dụng



Google Classroom



Visual Studio Code



# Nhóm công cụ giao tiếp



Google Classroom

Lưu trữ bài giảng, bài tập, nhận xét bài tập



Tham gia buổi học online qua Zoom



Chat community, chat riêng

# Nhóm công cụ coding



Quản lý code ở máy cá nhân



Đưa code lên online để trao đổi với đồng nghiệp

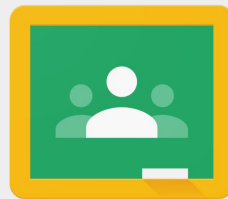


Visual Studio Code

Code editor, dùng để code cho đẹp và tiện

# Google classroom

Cách tìm học liệu, nộp bài



Google Classroom

# Công cụ học: Google Class

- Link của lớp học

Thông tin **khai giảng** khoá học Full-stack Automation QA với Playwright TypeScript K18 ➔ [Inbox x](#)

**Better Bytes Academy**  
to me, ngantq141001 ▾

Thân gửi: Học viên khóa K18 - Full-stack Automation QA từ chưa biết gì

Lời đầu tiên, trung tâm Better Bytes Academy - Playwright Việt Nam xin gửi tới các học viên lời chào và những lời chúc tốt đẹp gia học tập tại Better Bytes Academy.

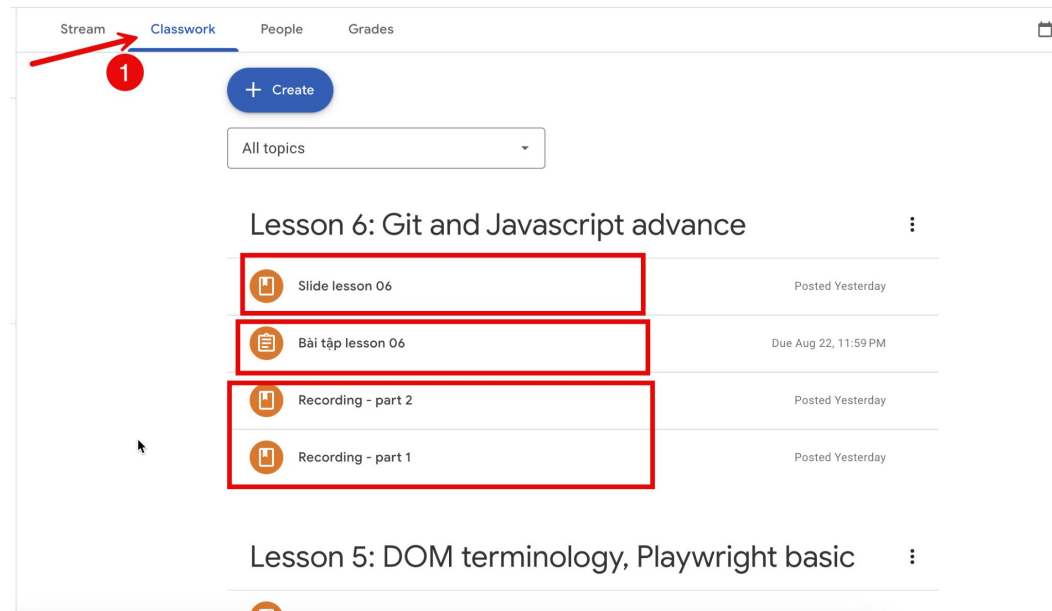
Trung tâm xin gửi các bạn một số thông tin chung của lớp như sau:

Thông tin lớp học:

|                                                                                           |                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tên lớp                                                                                   | K18                                                                                                                                                                    |
| Lịch học                                                                                  | Thứ 4 (20h30 - 22:00), Chủ nhật (20h30 - 22:00) GMT+7                                                                                                                  |
| Ngày <b>khai giảng</b>                                                                    | Thứ 4 (10/09/2025)                                                                                                                                                     |
| Group chat chung của lớp (học viên join để hỏi đáp và nhận thông tin trong suốt khoá học) | <a href="https://m.me/fj/AbYXLv2nZPqxZl0J/">https://m.me/fj/AbYXLv2nZPqxZl0J/</a>                                                                                      |
| Google Classroom (học viên join để xem tài liệu và nộp bài tập)                           | Class code: <b>gmvrcl3a</b><br><a href="https://classroom.google.com/c/ODAyMjc0Mzc4MzM1?cjc=gmvrcl3a">https://classroom.google.com/c/ODAyMjc0Mzc4MzM1?cjc=gmvrcl3a</a> |

# Công cụ học: Google Class

- Link của lớp học
- Cách xem lại slide, bài giảng, record



# Công cụ học: Google Class

- Link của lớp học
- Cách xem lại slide, bài giảng, record
- Cách nộp bài tập

Stream

1 Classwork

People

**Chọn Tab classroom** View your work

All topics

## Lesson 1



Lesson 1 - exercises

2

Due Aug 20, 11:59 PM

Posted 4:40 PM (Edited 4:42 PM)

Assigned

**Click vào bài tập và  
chọn instruction**

View instruction

3



## Lesson 1 - exercises

Playwright Việt Nam • 4:40 PM (Edited 4:42 PM)

100 points

Due Aug 20, 11:59 PM



Class comments

[Add a class comment](#)

**Chọn add or create và  
nhập link làm bài**

Your work 1 Assigned

[+ Add or create](#)

Google Drive

Link 2

File

Create new

Docs

Slides

Sheets

Drawings



## Lesson 1 - exercises

Playwright Việt Nam • 4:40 PM (Edited 4:42 PM)

100 points

Due Aug 20, 11:59 PM



Class comments

[Add a class comment](#)

**Bấm turn in để nộp bài**



### Your work

Assigned



Page not found · ...  
<https://github.com/playwright>



[+ Add or create](#)

Turn in

1



Private comments

[Add comment to Playwright Việt Nam](#)



# Bài tập

Nga Quỳnh • May 8, 2024

100 points



Lesson 1: Exercise 1

Google Docs



1 class comment



Phong Do 8:11 AM

Nhận xét bài tập:

Bài 1: cần sửa lại một chút



Add class comment...



**Nếu muốn chấm lại,  
un-submit và submit  
lại**

Your work

Turned in



Sign in - Google Account  
<https://classroom.google.com>

Unsubmit



Private comments

[Add comment to Nga Quỳnh](#)

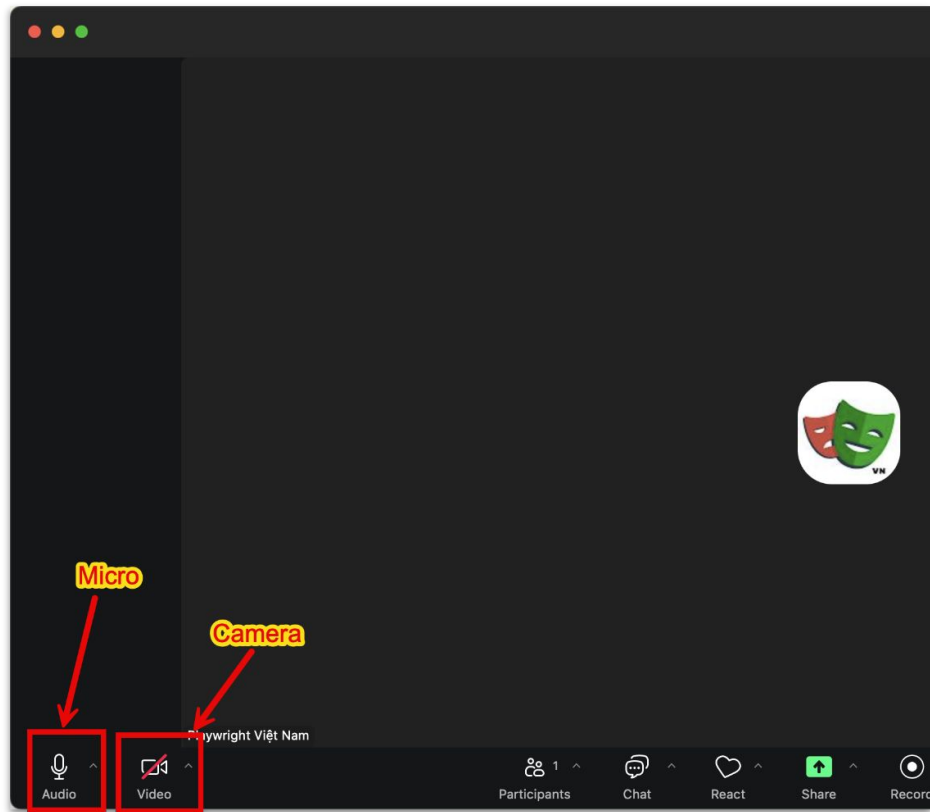
# Zoom

Sử dụng Zoom thế nào



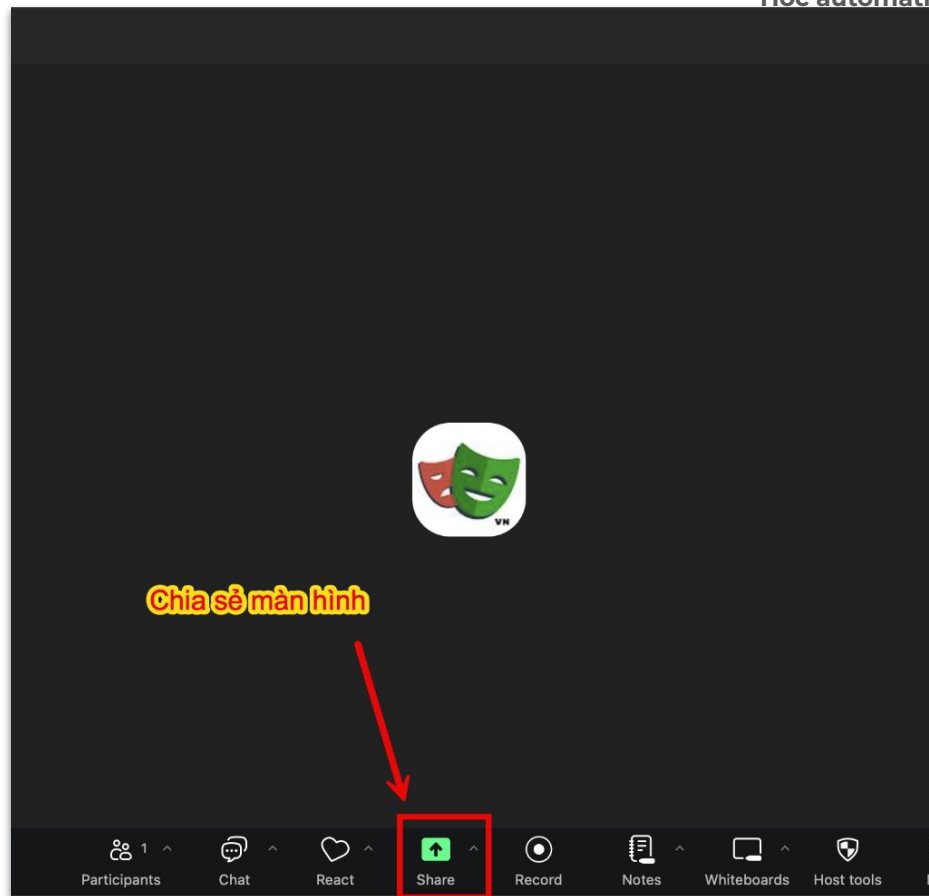
# Công cụ học: Zoom

- Micro và camera luôn tắt
  - Giữ sự riêng tư
  - Âm thanh tốt nhất cho lớp học
- Khi cần trao đổi, bạn có thể bật mic



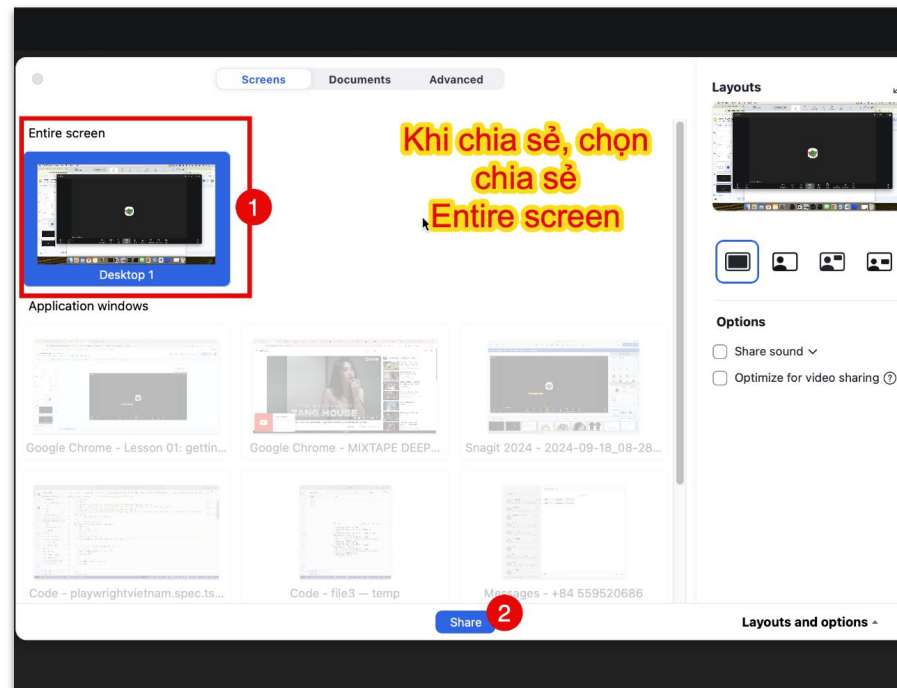
# Công cụ học: Zoom

- Chia sẻ màn hình:
  - Luôn chọn "entire screen" để mentor thấy toàn bộ các ứng dụng
    - VSCode, Playwright Browser,...



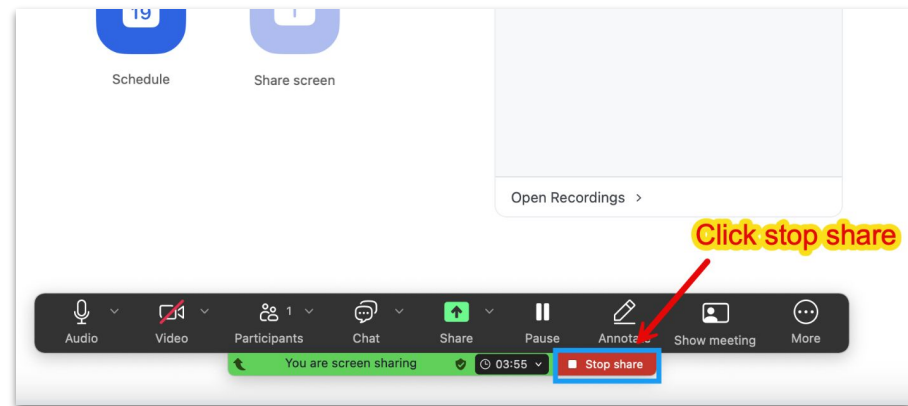
# Công cụ học: Zoom

- Chia sẻ màn hình:
  - Luôn chọn "entire screen" để mentor thấy toàn bộ các ứng dụng
    - VSCode, Playwright Browser,...



# Công cụ học: Zoom

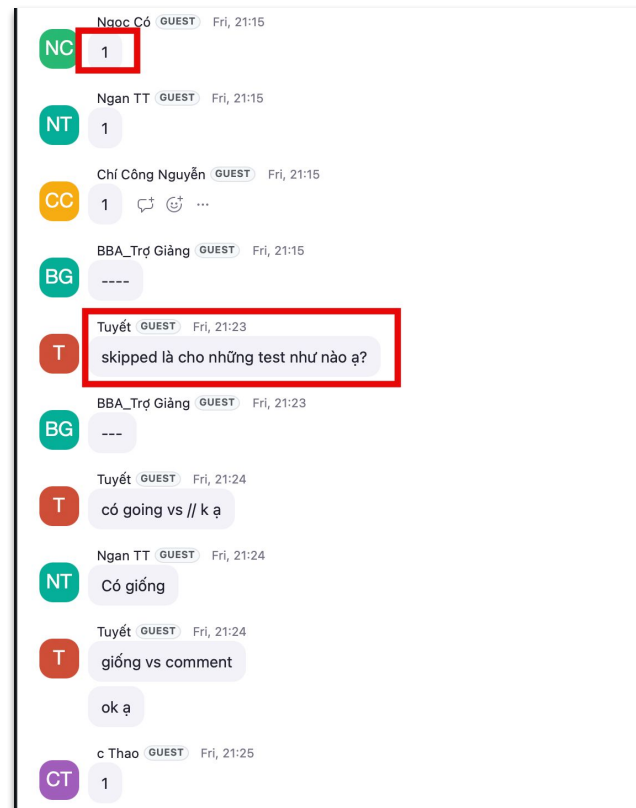
- Sau khi chia sẻ xong:
  - Hover xuống vùng phía dưới start menu
  - Click "stop share"



# Công cụ học: Zoom

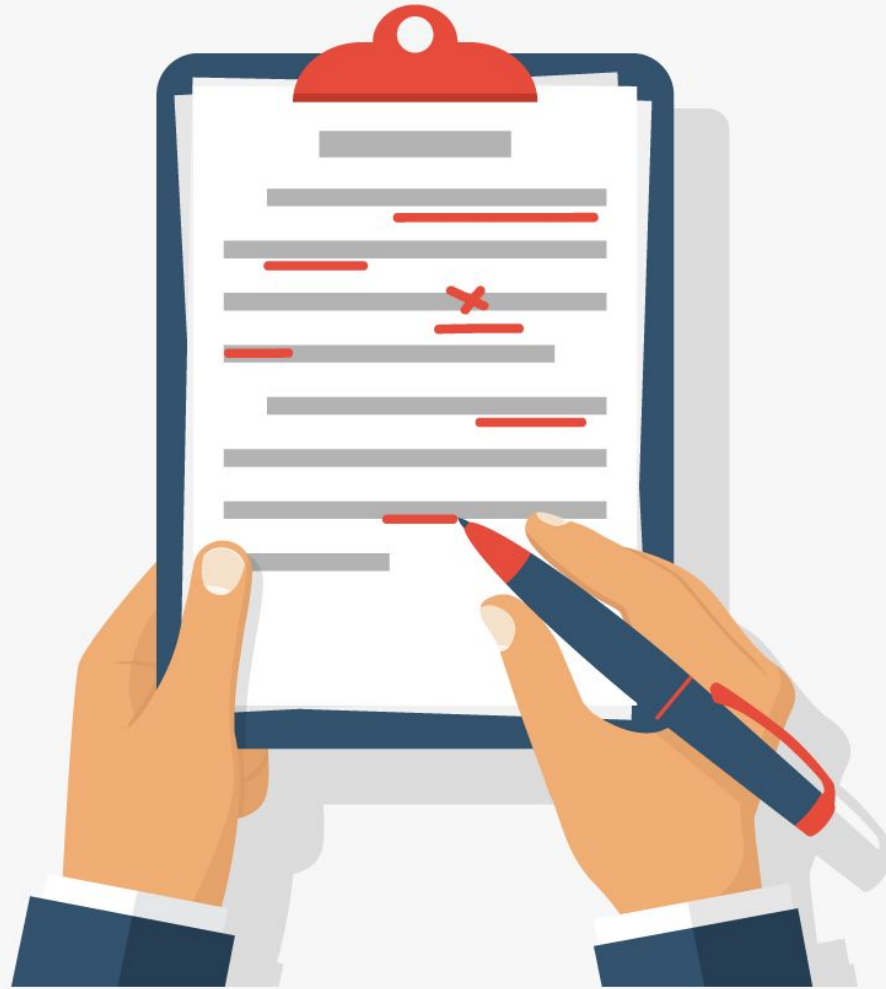
## Tương tác với lớp:

- Chat "1" nếu đã hiểu bài
- Có vấn đề:
  - Bật mic trao đổi (cách tốt nhất)
  - Chat hỏi trong box chat



# Tài liệu học

Tài liệu học tập tại  
Playwright Việt Nam



# Tài liệu học: truy cập trọn đời

- Các tài liệu học, học viên được truy cập trọn đời
  - Slide
  - Record
  - Bài tập



# Tài liệu học: private

- Vui lòng không share dưới mọi hình thức
  - Học chung
  - Copy lại về driver cá nhân
  - Share recording
  - ...



# Tài liệu học: hình thức xử lý vi phạm

- Với các trường hợp vi phạm:
  - Dừng hợp tác với học viên, dừng hỗ trợ
  - Remove access vào tài liệu, không hoàn tiền
  - Không thể đăng kí các khóa học khác từ trung tâm



# Hợp tác lâu dài, bền vững

Đây là các tài liệu, tâm huyết từ  
phía trung tâm

Rất mong học viên tôn trọng và  
xây dựng mối **quan hệ bền vững**  
với trung tâm ^^



# Playwright

Playwright là gì?  
Tại sao chọn Playwright  
TypeScript?



# Playwright là gì?



- Là một **framework**
- Tiền thân là **Puppeteer**, được Microsoft tài trợ và phát triển lên.
- <https://playwright.dev/>
- Ra đời 31/01/2020
- Version mới nhất: **v1.56**



# Playwright là gì?



## Ưu điểm Playwright

- **Cross browser**
  - Hỗ trợ các trình duyệt phổ biến: Chrome, Edge, Firefox, Safari
  - Hỗ trợ lên tới 143 phiên bản trình duyệt



```
type Devices = {  
  "Blackberry PlayBook": DeviceDescriptor;  
  "Blackberry PlayBook landscape": DeviceDescriptor;  
  "BlackBerry Z30": DeviceDescriptor;  
  "BlackBerry Z30 landscape": DeviceDescriptor;  
  "Galaxy Note 3": DeviceDescriptor;  
  "Galaxy Note 3 landscape": DeviceDescriptor;  
  "Galaxy Note II": DeviceDescriptor;  
  "Galaxy Note II landscape": DeviceDescriptor;  
  "Galaxy S III": DeviceDescriptor;  
  "Galaxy S III landscape": DeviceDescriptor;  
  "Galaxy S5": DeviceDescriptor;  
  "Galaxy S5 landscape": DeviceDescriptor;  
  "Galaxy S8": DeviceDescriptor;  
  "Galaxy S8 landscape": DeviceDescriptor;  
  "Galaxy S9+": DeviceDescriptor;  
  "Galaxy S9+ landscape": DeviceDescriptor;  
  "Galaxy S24": DeviceDescriptor;  
  "Galaxy S24 landscape": DeviceDescriptor;  
}
```

# Playwright là gì?



## Ưu điểm Playwright

- Cross browser
- **Cross platform**
  - Code một lần, chạy trên các hệ điều hành phổ biến: Windows, Linux, MacOS



# Playwright là gì?



## Ưu điểm Playwright

- Cross browser
- Cross platform
- **Tính năng “xịn sò”**
  - auto waiting, auto-retry  
assertion giúp **giảm flaky tests** (lúc pass, lúc fail)

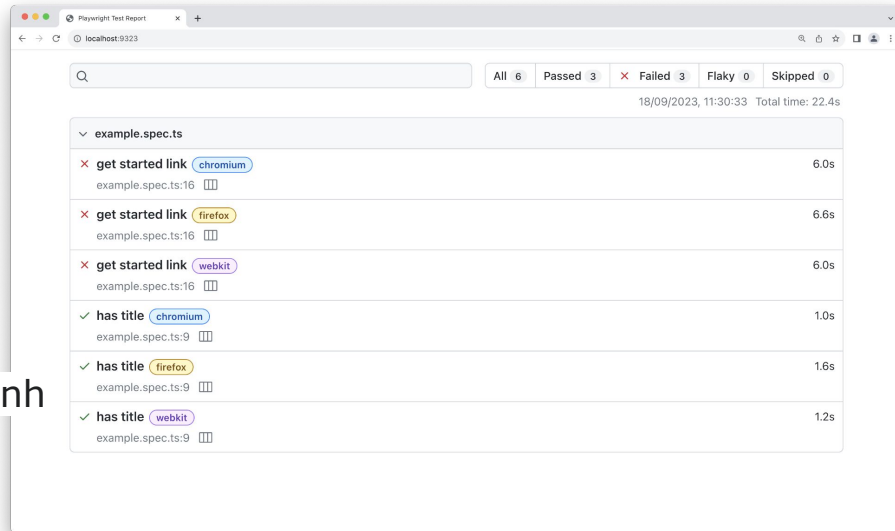
```
© Log
waiting for locator('//button[@data-product-id=\'1\']') 13ms
locator resolved to <button class="add-to-cart" data-product-id="1">Add to Cart</button> 1ms
attempting click action 0ms
waiting for element to be visible, enabled and stable 15ms
element is visible, enabled and stable 0ms
scrolling into view if needed 0ms
done scrolling 6ms
performing click action 33ms
click action done 0ms
waiting for scheduled navigations to finish 0ms
navigations have finished 0ms
```

# Playwright là gì?



## Ưu điểm Playwright

- Cross browser
- Cross platform
- Tính năng "xịn sò"
- **Report đầy đủ thông tin**
  - Pass/fail theo từng loại trình duyệt
  - Chi tiết ở từng thời điểm: "gọi API nào", "response trả về gì", "ứng với dòng code nào"



Playwright Test Report

localhost:9323

All 6 Passed 3 Failed 3 Flaky 0 Skipped 0

18/09/2023, 11:30:33 Total time: 22.4s

example.spec.ts

|                    |          |      |
|--------------------|----------|------|
| ✗ get started link | chromium | 6.0s |
| ✗ get started link | firefox  | 6.6s |
| ✗ get started link | webkit   | 6.0s |
| ✓ has title        | chromium | 1.0s |
| ✓ has title        | firefox  | 1.6s |
| ✓ has title        | webkit   | 1.2s |

Hiển thị pass/fail rõ ràng theo từng loại trình duyệt

Playwright Test Report

localhost:9323/#?testId=a30a6eba6312f6b87ea5-274e97907a6e3638a22f

Search

All 6 Passed 3 Failed 3 Flaky 0 Skipped 0

get started link

example.spec.ts:16 6.0s

chromium

Run

Errors

Test Steps

Traces

trace

**Click vào từng test, hiển thị chi tiết từng test**

Playwright Trace Viewer

localhost:9323/trace/index.html?trace=http://localhost:9323/data/8672e3b2631ec8ee126126a49bca107a681220d4.zip

example.spec.ts:16 > get started link

0 500ms 1.0s 1.5s 2.0s 2.5s 3.0s 3.5s 4.0s 4.5s 5.0s 5.5s 6.0s

Actions Metadata

> Before Hooks 737ms

page.goto https://playwright... 81ms

locator.click getByRole('link'... 70ms

expect.toBeVisible getByRole... 5.1s

> After Hooks 1ms

Action Before After

Installation

Getting Started Docs API Node.js Community

Getting Started Installation

Installation

Installing Playwright

Timed out 5000ms waiting for expect(received).toBeVisible()

Call log:

- expect.toBeVisible with timeout 5000ms
- waiting for getByRole('heading', { name: 'get started' })

908ms expect.toBeVisible getByRole('heading', { name: 'get started' }) 5.1s

Locator Call Log Errors 1 Console Network 93 Source Attachments

Trace hiển thị chi tiết:  
Thời gian chạy, lỗi ở  
đâu, dòng nào

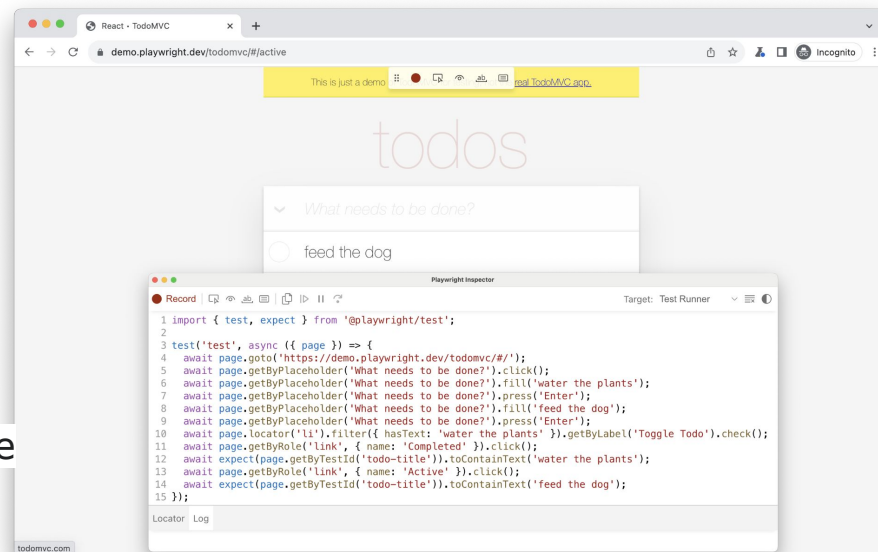
ation test  
biết gì

# Playwright là gì?



## Ưu điểm Playwright

- Cross browser
- Cross platform
- Tính năng "xịn sò"
- Report đầy đủ thông tin
- **Code gen**
  - Thao tác để sinh ra code

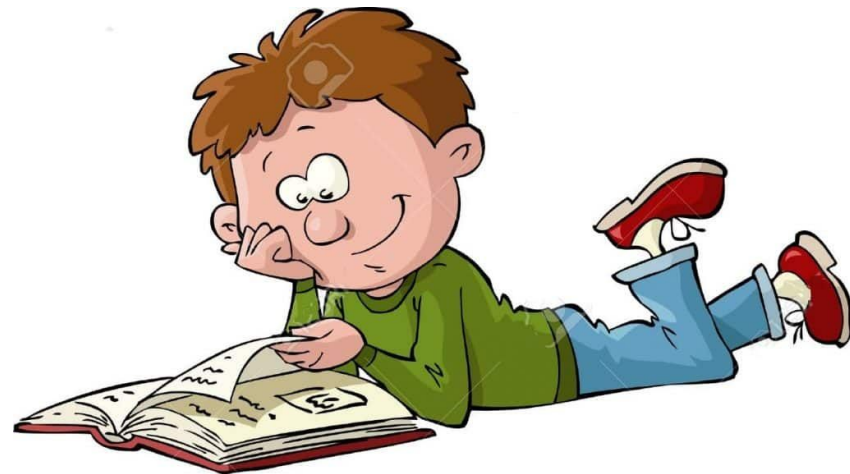


# Tại sao chọn học Playwright TypeScript?



## Học Playwright vì:

- Dễ cài đặt
- Cú pháp đơn giản, dễ học
- Framework trending, nhiều cơ hội việc làm.



# Tại sao chọn học Playwright TypeScript?

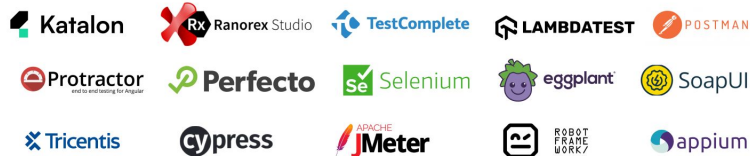


Công ty dùng **tool khác** thì có nên học **khoá này không?**

**Có!** Vì:

- Khóa học dạy về tư duy, hiểu bản chất
- Hiểu bản chất thì các tools tương tự nhau
- Mentor có kiến thức đa dạng các tools, có thể hỏi về tools của công ty, mentor sẽ hỗ trợ

## Top 15 automation testing tools



# Giải thích các công cụ đã cài trước buổi học



# NVM



**NVM** = Node Version Manager = quản  
lý các phiên bản NodeJs

NodeJs = Công cụ để chạy code



# NVM



**NVM** = Node Version Manager = quản lý các phiên bản NodeJs

NodeJs = Công cụ để chạy code

**2 options:**

- Cài trực tiếp NodeJS vào máy
- Cài thông qua NVM



# NVM

# NVM



**NVM** = Node Version Manager = quản lý các phiên bản NodeJs

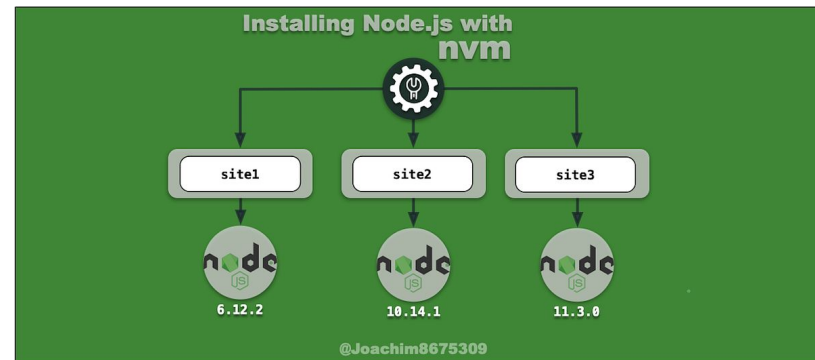
NodeJs = Công cụ để chạy code

## 2 options:

- Cài trực tiếp NodeJS vào máy
- Cài thông qua NVM

**Chọn thông qua NVM** vì

- Dễ chuyển đổi nhiều phiên bản NodeJS (các dự án khác nhau có thể làm ở phiên bản NodeJS khác nhau)



# Git & GitHub



**Git:** quản lý source code

**GitHub:** chia sẻ code, làm việc nhóm



# Git & GitHub



**Thực hành:** Submit tài khoản  
GitHub



[https://go.betterbytesvn.com/  
BBA-student-submit-GitHub](https://go.betterbytesvn.com/BBA-student-submit-GitHub)

Submit tài khoản GitHub

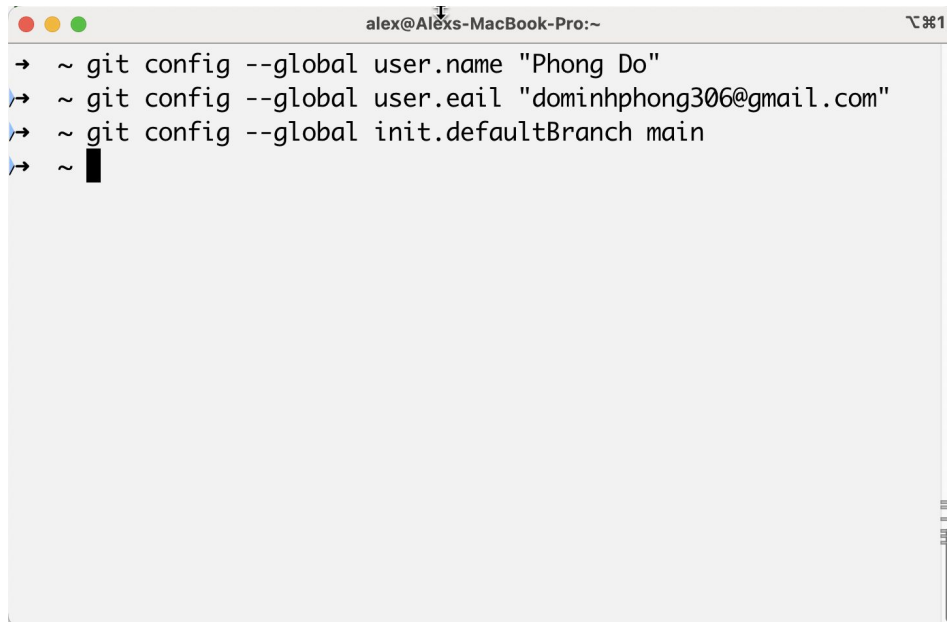
# Cấu hình Git



Trước khi làm việc với Git, cần một số **cấu hình mặc định**:

- Config **username** (tên người dùng):
  - `git config --global user.name "<tên bạn>"`
- Config **email** (địa chỉ email):
  - `git config --global user.email "<email của bạn>"`
- Config **branch default** (nhánh mặc định):
  - `git config --global init.defaultBranch main`

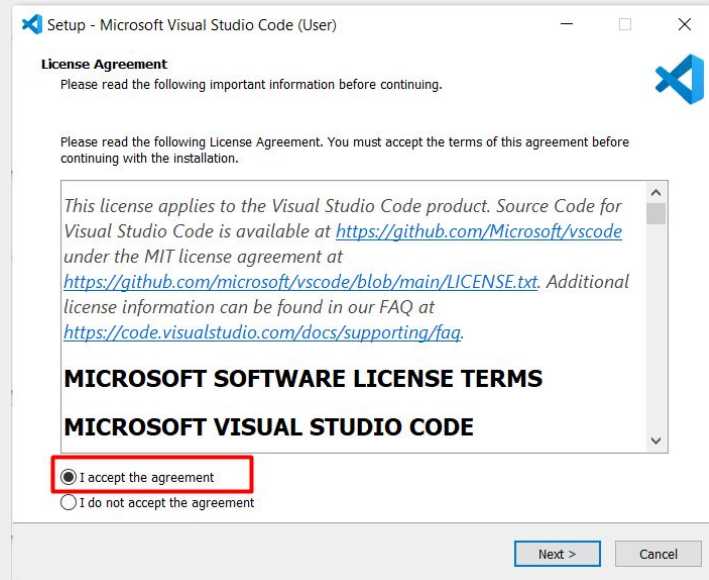
# Cài đặt: Cấu hình git

A screenshot of a macOS terminal window titled 'alex@Alexs-MacBook-Pro:~'. The window shows a series of four git configuration commands being entered, each preceded by a blue arrow icon. The commands are: 'git config --global user.name "Phong Do"', 'git config --global user.eail "dominhphong306@gmail.com"', 'git config --global init.defaultBranch main', and a final prompt '~' with a black cursor. The terminal has a light gray background and standard macOS window controls at the top.

```
alex@Alexs-MacBook-Pro:~  
→ ~ git config --global user.name "Phong Do"  
→ ~ git config --global user.eail "dominhphong306@gmail.com"  
→ ~ git config --global init.defaultBranch main  
→ ~
```

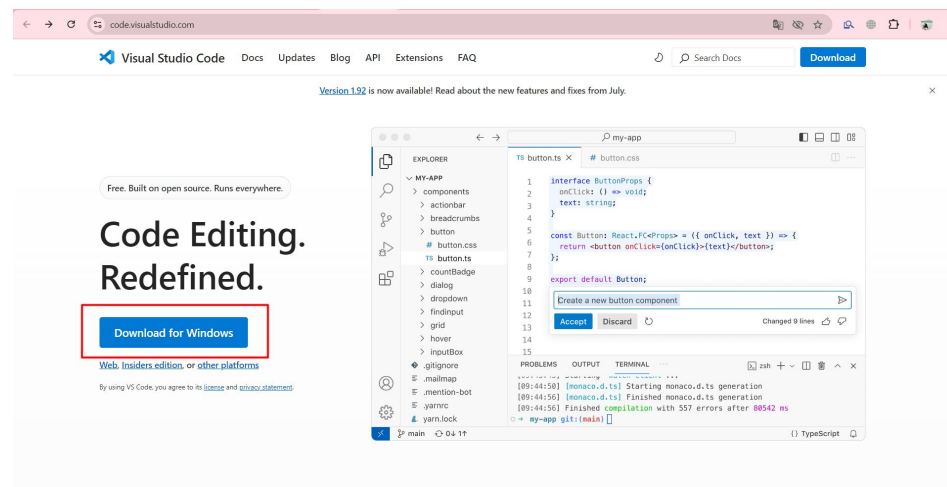
# Cài đặt tools

VS Code



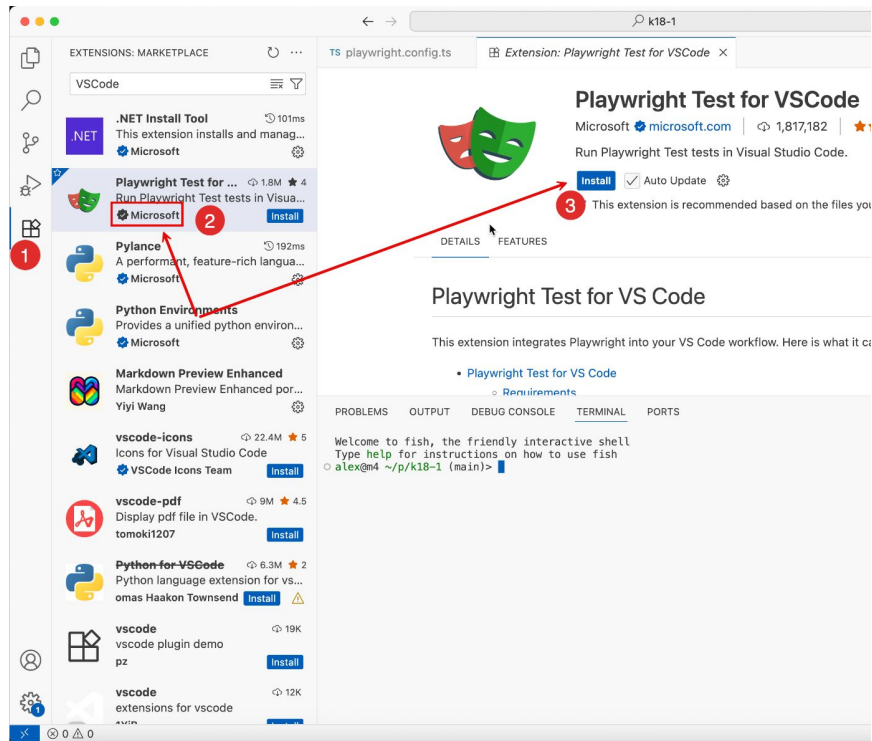
# Cài đặt: Visual Studio Code

- VS Code = **IDE** = integrated development environment
  - **Là công cụ để viết code**
- Cài đặt Visual Studio Code:  
<https://code.visualstudio.com/>



# Cài đặt: Visual Studio Code

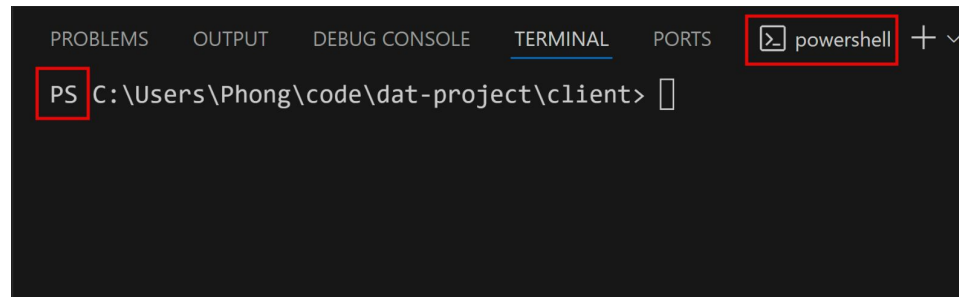
- VS Code = **IDE** = integrated development environment
  - Là công cụ để viết code**
- Cài đặt Visual Studio Code: <https://code.visualstudio.com/>
- Cài đặt extension
  - Playwright extension:** <https://marketplace.visualstudio.com/items?itemName=ms-playwright.playwright>



# Cài đặt: Visual Studio Code

## Đổi terminal mặc định

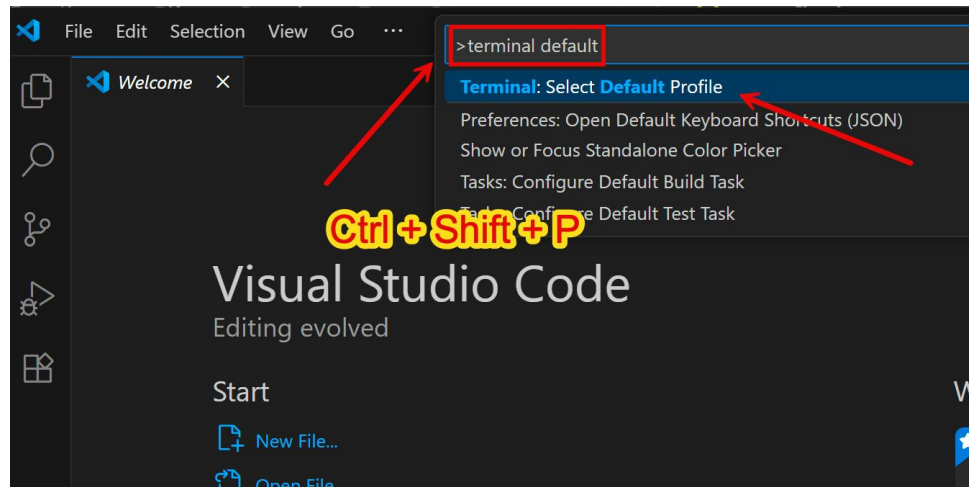
- Window Powershell là terminal mặc định trên Windows
- Powershell hay bị chặn/ lỗi lặt vặt
- => dùng **Git Bash** để tránh các lỗi này



# Cài đặt: Visual Studio Code

## Đổi terminal mặc định

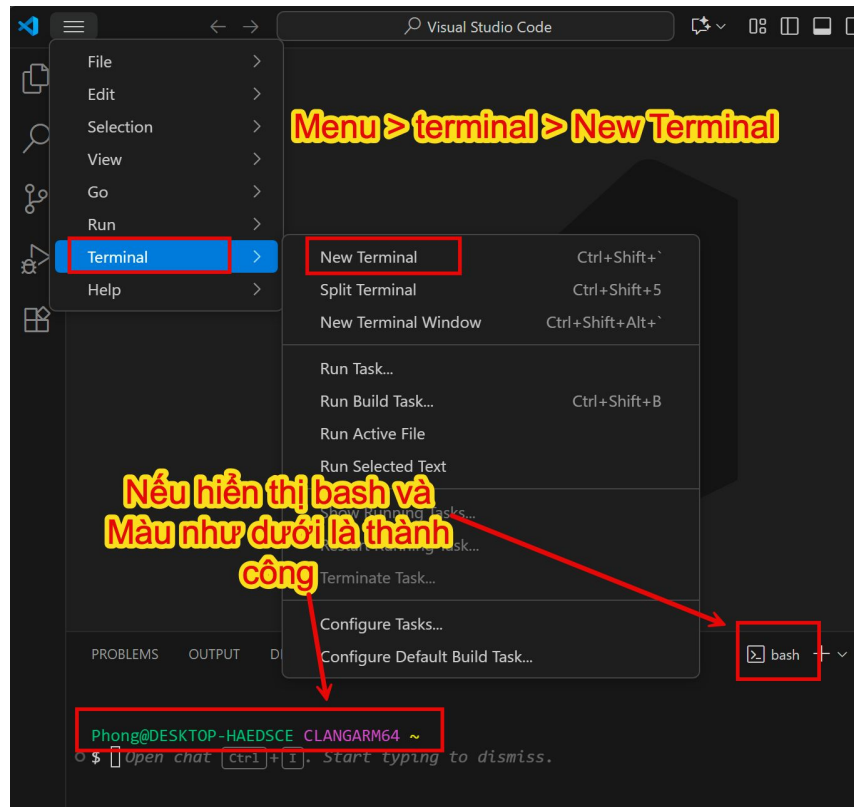
- **Ctrl + Shift + P**: Hiện thị hộp thoại
- **Tìm kiếm**: Terminal default
- **Chọn**:  
Terminal: Select Default Profile



# Cài đặt: Visual Studio Code

## Đổi terminal mặc định

- **Ctrl + Shift + P**: Hiển thị hộp thoại
- **Tìm kiếm**: Terminal default
- **Chọn**:  
Terminal: Select Default Profile
- Chọn Git Bash
- Kiểm tra lại



# Kết nối với GitHub

Tạo SSH key  
Thêm SSH key vào GitHub



# Kết nối với GitHub

Local computer



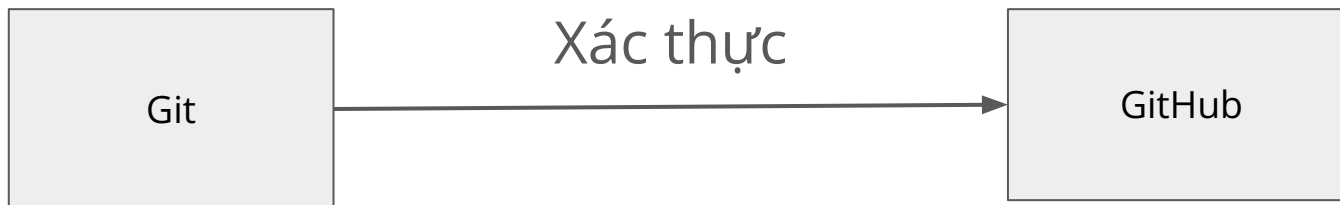
Online



# Kết nối với GitHub

Local computer

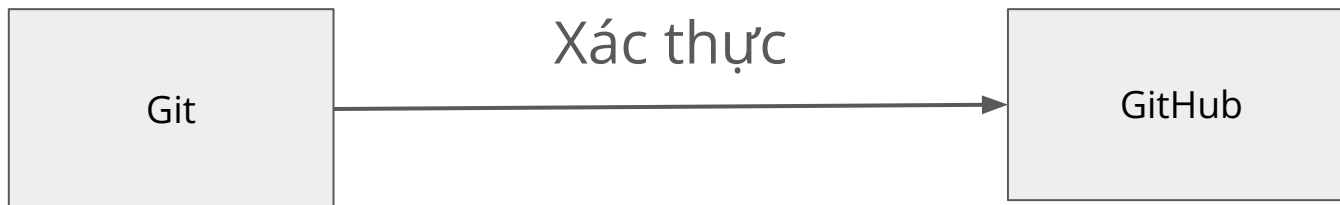
Online



# Kết nối với GitHub

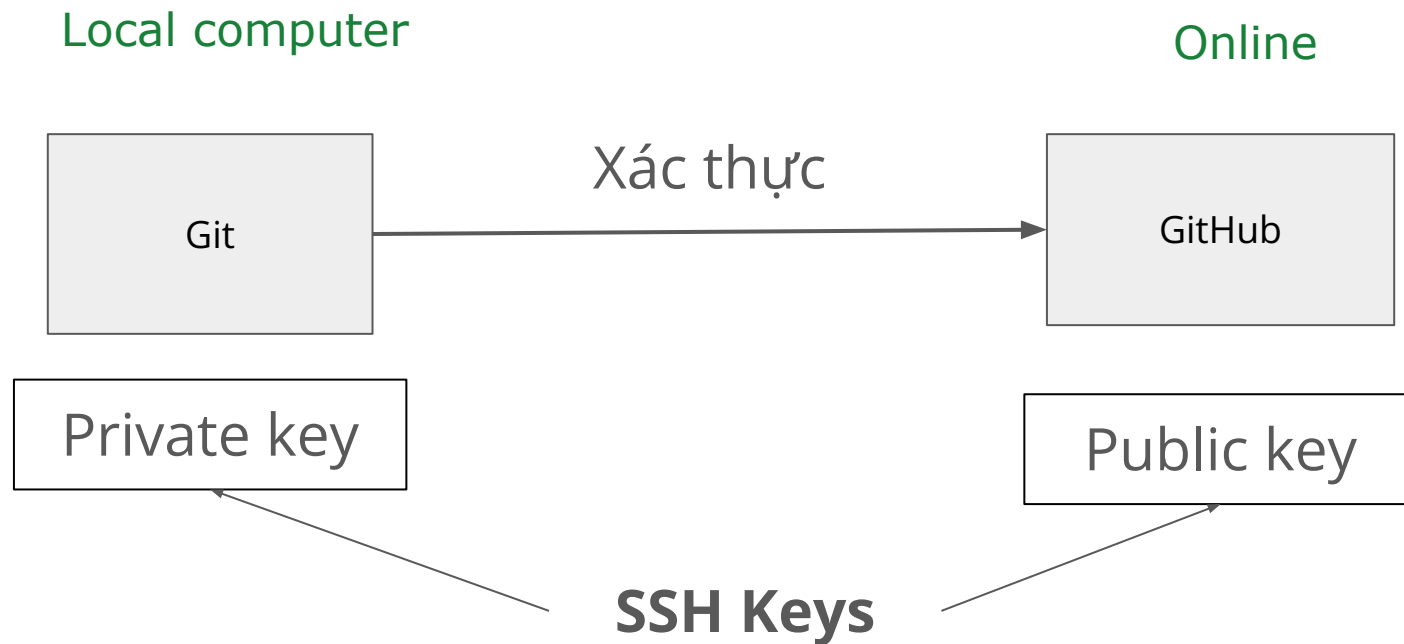
Local computer

Online

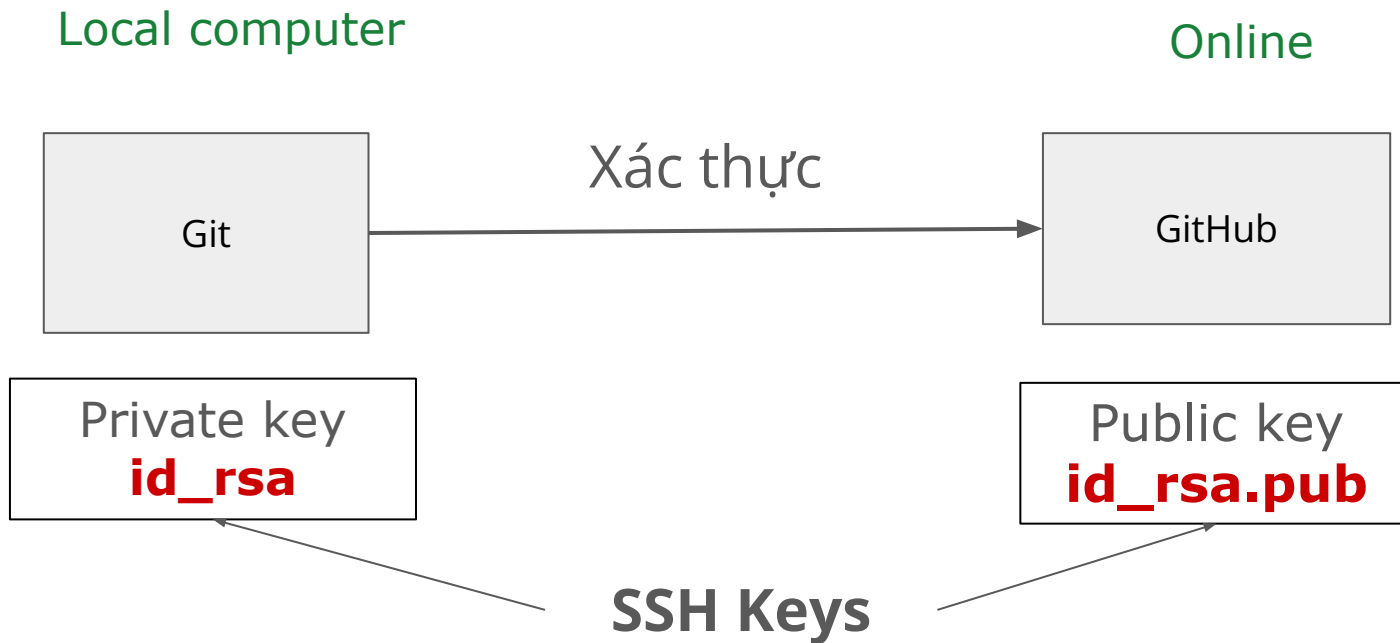


SSH Keys

# Kết nối với GitHub

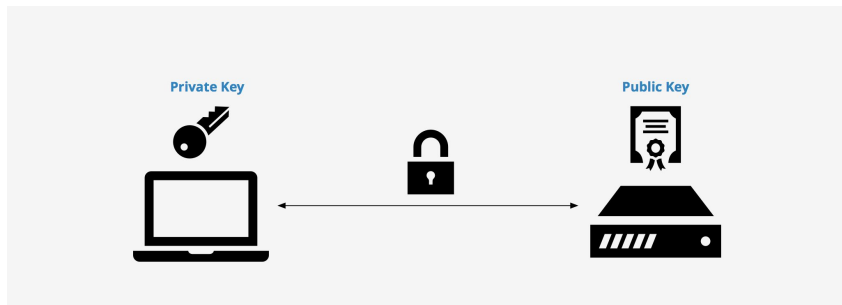


# Kết nối với GitHub



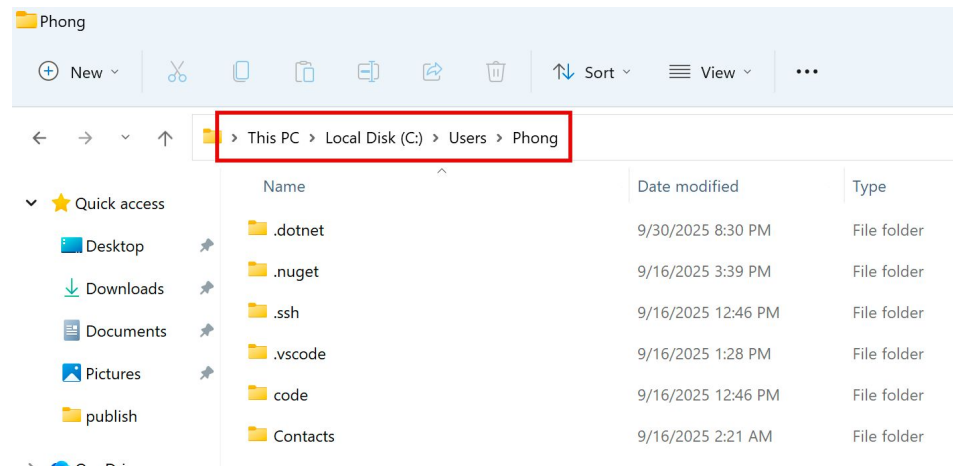
# Kết nối GitHub: SSH Key là gì?

- SSH key:
  - **Cặp** khoá (2 cái)
    - `id_rsa` và `id_rsa.pub`
    - `id_rsa`: cần giữ bí mật
    - `id_rsa.pub`: có thể gửi cho người khác
  - Giúp **xác thực** đăng nhập trở nên **dễ dàng hơn**
  - Lưu ở `~/.ssh`
  - "`~`" đại diện cho thư mục home



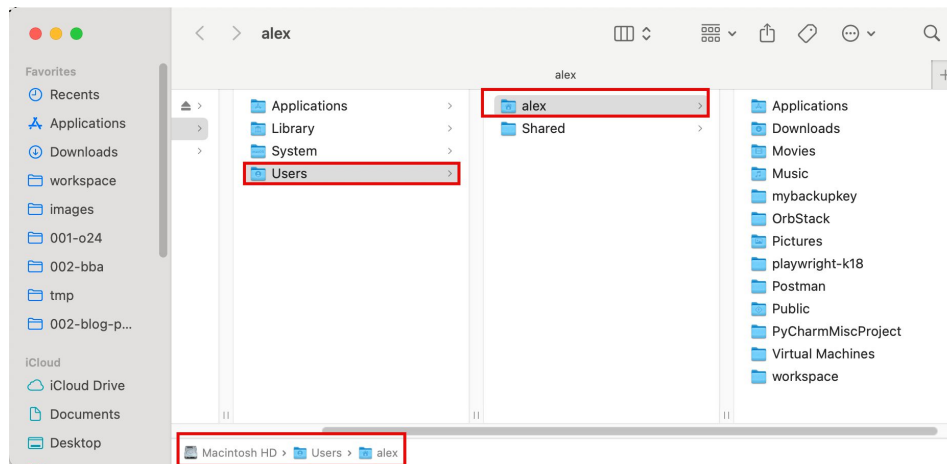
# Kết nối GitHub: SSH Key là gì?

- **Home** ở Windows:
  - `C:\Users\{username}`



# Kết nối GitHub: SSH Key là gì?

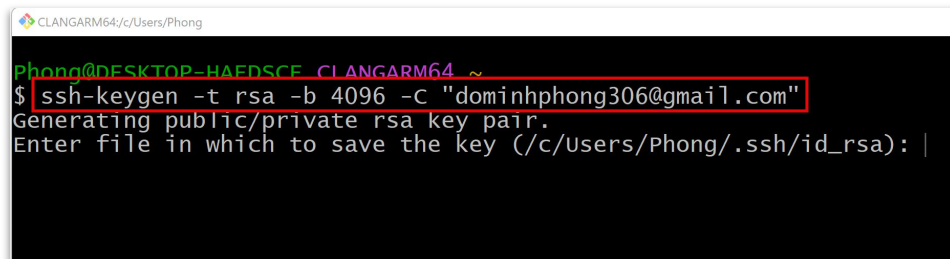
- **Home** ở Windows:
  - `C:\Users\{username}`
- **Home** ở Linux/ MacOS
  - `/Users/{username}`



# Kết nối GitHub: Tạo SSH key

## Lệnh tạo SSH Keys:

```
ssh-keygen -t rsa -b 4096 -C  
"your_email@example.com"
```

A terminal window with a black background and white text. The title bar at the top reads 'CLANGARM64/c/Users/Phong'. The prompt is 'Phong@DESKTOP-HAFDSCF CLANGARM64 ~'. The command '\$ ssh-keygen -t rsa -b 4096 -C "dominhphong306@gmail.com"' is entered and highlighted with a red box. The output shows 'Generating public/private rsa key pair.' and 'Enter file in which to save the key (/c/Users/Phong/.ssh/id\_rsa): |'.

```
CLANGARM64/c/Users/Phong  
Phong@DESKTOP-HAFDSCF CLANGARM64 ~  
$ ssh-keygen -t rsa -b 4096 -C "dominhphong306@gmail.com"  
Generating public/private rsa key pair.  
Enter file in which to save the key (/c/Users/Phong/.ssh/id_rsa): |
```

Phong@DESKTOP-HAEDSCE CLANGARM64 ~

\$ ssh-keygen -t rsa -b 4096 -C "dominhphong306@gmail.com"

**Câu lệnh gen key**

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/Phong/.ssh/id\_rsa):

Enter passphrase for "/c/Users/Phong/.ssh/id\_rsa" (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/Phong/.ssh/id\_rsa

Your public key has been saved in /c/Users/Phong/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:BHcLoA9nyA4qm+0q7fktg+J8/yeginHQ32SE5c3roJM dominhphong306@gmail.com

The key's randomart image is:

+----[RSA 4096]-----+

```
|      o+.o  .  
|    ..+. = o .  
|  .. =.+ + .  
| ...o =o. .  
| oo ..+. S  
| o+. ....  
| +oo. . .  
| =oo+o.Eo .  
| +*=.o++=o  
|
```

+----[SHA256]-----+

**Vị trí lưu mặc định  
Không cần thay đổi**

Phong@DESKTOP-HAEDSCE CLANGARM64 ~

\$ ssh-keygen -t rsa -b 4096 -C "dominhphong306@gmail.com"

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/Phong/.ssh/id\_rsa):

Enter passphrase for "/c/Users/Phong/.ssh/id\_rsa" (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/Phong/.ssh/id\_rsa

Your public key has been saved in /c/Users/Phong/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:BHcLoA9nyA4qm+0q7fktg+J8/yeginHQ32SE5c3roJM dominhphong306@gmail.com

The key's randomart image is:

+---[RSA 4096]-----+

```
|      O+.O  .  
|    ..+.= O  .  
|   .. =.+ +  .  
|  ...O =O.  .  
| oo  ..+. S  
| O+.  ....  
| +OO..  . .  
| =OO+O.EO  .  
| +*=.O++=O
```

+-----[SHA256]-----+

Mật khẩu của key  
Bỏ trống luôn

Phong@DESKTOP-HAEDSCE CLANGARM64 ~

\$ ssh-keygen -t rsa -b 4096 -C "dominhphong306@gmail.com"

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/Phong/.ssh/id\_rsa):

Enter passphrase for "/c/Users/Phong/.ssh/id\_rsa" (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/Phong/.ssh/id\_rsa

Your public key has been saved in /c/Users/Phong/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:BHcLoA9nyA4qm+0q7fktg+J8/yeginHQ32SE5c3rOjM dominhphong306@gmail.com

The key's randomart image is:

+---[RSA 4096]-----+

```
|      O+.O  .  
|    ..+.= O  .  
|   .. =.+ +  .  
|  ...O =O.  .  
| oo  ..+. S  
| O+. ....  
| +OO..  . .  
| =OO+O.EO  .  
| +*=.O++=O  
|
```

+-----[SHA256]-----+

← **Hiển thị thông báo  
thể này là thành  
công**

Phong@DESKTOP-HAEDSCE CLANGARM64 ~

```
$ ssh-keygen -t rsa -b 4096 -C "dominhphong306@gmail.com"
```

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/Phong/.ssh/id\_rsa):

/c/Users/Phong/.ssh/id\_rsa **already exists.**

Overwrite (y/n)? n

Phong@DESKTOP-HAEDSCE CLANGARM64 ~

\$

**Nếu đã có key rồi thì không cần gen lại. Dừng lại luôn.**

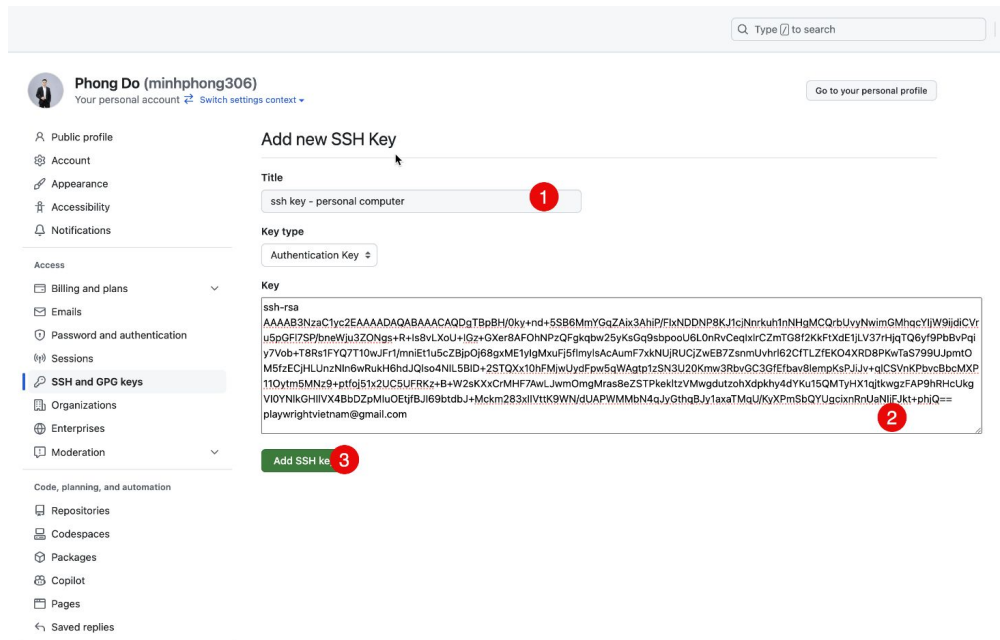
# Kết nối SSH: thêm vào GitHub

- Lấy nội dung ssh key: **cat**  
**~/.ssh/id\_rsa.pub**

```
→ ~ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDgTBpBH/0ky+nd+5SB6MmYGqZAix3
AhiP/FIXNDDNP8KJ1cjNnrkuh1nNHgMCQrbUvyNwimGMhacYIjW9ijdiCVru5pGF17S
P/bneWju3ZONgs+R+Is8vLXuU+IGz+GXer8AF0hNPzQFgkqbw25yKsGq9sbpooU6L0n
RvCeqIxlrCZmTG8f2KkFtXdE1jLV37rHjqTQ6yf9PbBvPqiY7Vob+T8Rs1FYQ7T10wJ
Fr1/mniEt1u5cZBjp0j68gxME1yIgMxufj5flmylsAcAumF7xkNUjRUCjZwEB7ZsnmU
vhr162CfTLZfEK04XRD8PKwTaS799UJpmtOM5fzECjHLUnzNI6wRukH6hdJQ1so4NI
L5BID+2STQXx10hFMjwUydFpw5qWAgtp1zSN3U20Kmw3RbvGC3GfEfBav8lEmpKsPji
Jv+qLCSVnKPbvcBbcMXP110ytm5MNz9+ptfoj51x2UC5UFRKz+B+W2sKXxCrMHF7AwL
Jwm0mgMras8eZSTPkekltzVMwgduztzohXdpkhy4dYKu15QMTyHX1qjtkwgzFAP9hRHc
UkgVI0YNlkGHILVX4BbDZpMIu0EtjfBJI69btdbJ+Mckm283xIIVttK9WN/dUAPWMMb
N4qJyGthqBJy1axaTMqU/KyXPmSbQYUgcixnRnUaNIjFJkt+phjQ== playwrightvi
etnam@gmail.com
→ ~
```

# Kết nối SSH: thêm vào GitHub

- Lấy nội dung ssh key: **cat**  
**~/.ssh/id\_rsa.pub**
- Truy cập:  
<https://github.com/settings/ssh/new> để thêm ssh key





**Phong Do (minhphong306)**

Your personal account [Switch settings context](#)

[Go to your personal profile](#)

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

**SSH and GPG keys**

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

## Add new SSH Key

Title

ssh key - personal computer

Key type

Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDgTBpBH/0ky+nd+5SB6MmYGqZAix3AhiP/FixNDDNP8KJ1cjNnrkuh1nNHgMCQrbUvyNwimGMhqcYIjW9iidiCvr
u5pGFI7SP/bneWju3ZONgs+R+Is8vLXoU+IGz+GXer8AFOhNPzQFGkqbw25yKsGq9sbpooU6L0nRvCeqIxlrCZmTG8f2KkFtXdE1jLV37rHjqTQ6yf9PbBvPqi
y7Vob+T8Rs1FYQ7T10wJFr1/mniEt1u5cZBjpOj68gxME1ylgMxuFj5flmylsAcAumF7xkNUjRUCJzWzEB7ZsnmUvhl62CfTLZfEKO4XRD8PKwTaS799UJpmtO
M5fzECjHLUnzNln6wRukH6hdJQIso4NIL5BID+2STQXx10hFMjwUydFpw5qWAgtP1zSN3U20Kmw3RbvGC3GfEfbaV8lempKsPjJjv+qICSVnKPbvcBbcMXP
11Oym5MNz9+ptfoj51x2UC5UFRKz+B+W2sKXxCrMHF7AwLJwmOmgMras8eZSTPkekltzVMwgduztzohXdpkhy4dYKu15QMTyHX1qjtkwgzFAP9hRHcUkg
V10YNikGHlVX4BbDZpMluOEtjBJI69btDbJ+Mckm283xliVttK9WN/dUAPWMMbN4qJyGthqBJy1axaTMqU/KyXPmSbQYUgcixnRnUaNIIFJkt+phjQ==
playwrightvietnam@gmail.com
```

Add SSH key

# Cài đặt Playwright

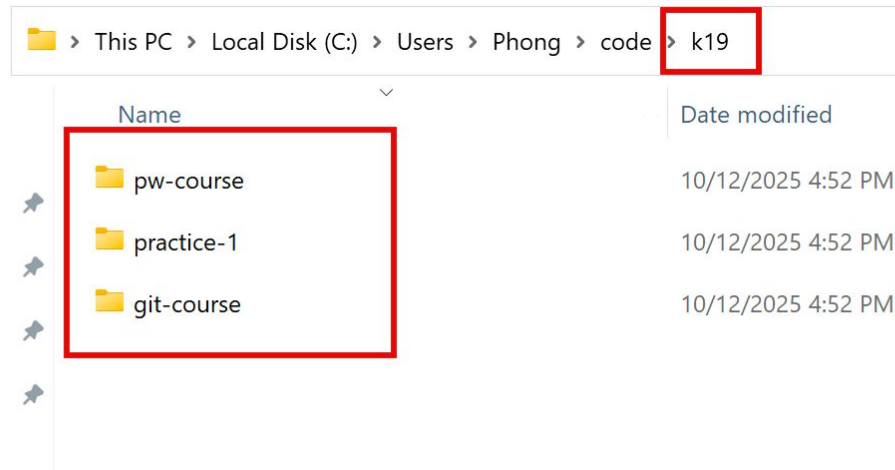
Chạy test đầu tiên



# Trước khi khởi tạo



Tạo một thư mục chứa code  
của toàn bộ khoá học  
Như thế gon gàng và dễ tìm  
kiếm hơn nhiều!



# Cài đặt Playwright

- Tạo thư mục: **demo-1**
- Mở thư mục bằng VS Code
- Mở terminal lên
- Chạy lệnh:
  - `npm init playwright@latest`
  - Liên tục gõ "enter" cho tới khi hiển thị như hình bên

```
alex@m4 ~/w/t/p/pw-course> npm init playwright@latest
Need to install the following packages:
create-playwright@1.17.139
Ok to proceed? (y) y

> tmp@1.0.0 npx
> create-playwright

Getting started with writing end-to-end tests with Playwright:
Initializing project in '.'
✓ Do you want to use TypeScript or JavaScript? · TypeScript
✓ Where to put your end-to-end tests? · tests
✓ Add a GitHub Actions workflow? (Y/n) · true
✓ Install Playwright browsers (can be done manually via 'npx playwright install')? (Y/n) · true
Initializing NPM project (npm init -y)...
Wrote to /Users/alex/workspace/tmp/pw/pw-course/package.json:
```

```
alex@m4 ~/w/t/p/pw-course> npm init playwright@latest
```

```
found 0 vulnerabilities
Writing playwright.config.ts.
Writing .github/workflows/playwright.yml.
Writing tests/example.spec.ts.
Writing package.json.
Downloading browsers (npx playwright install)...
✓ Success! Created a Playwright Test project at /Users/alex/workspace/tmp/pw/pw-course
```

Inside that directory, you can run several commands:

```
npx playwright test
  Runs the end-to-end tests.
```

```
npx playwright test --ui
  Starts the interactive UI mode.
```

```
npx playwright test --project=chromium
  Runs the tests only on Desktop Chrome.
```

```
npx playwright test example
  Runs the tests in a specific file.
```

```
npx playwright test --debug
  Runs the tests in debug mode.
```

```
npx playwright codegen
  Auto generate tests with Codegen.
```

We suggest that you begin by typing:

```
npx playwright test
```

And check out the following files:

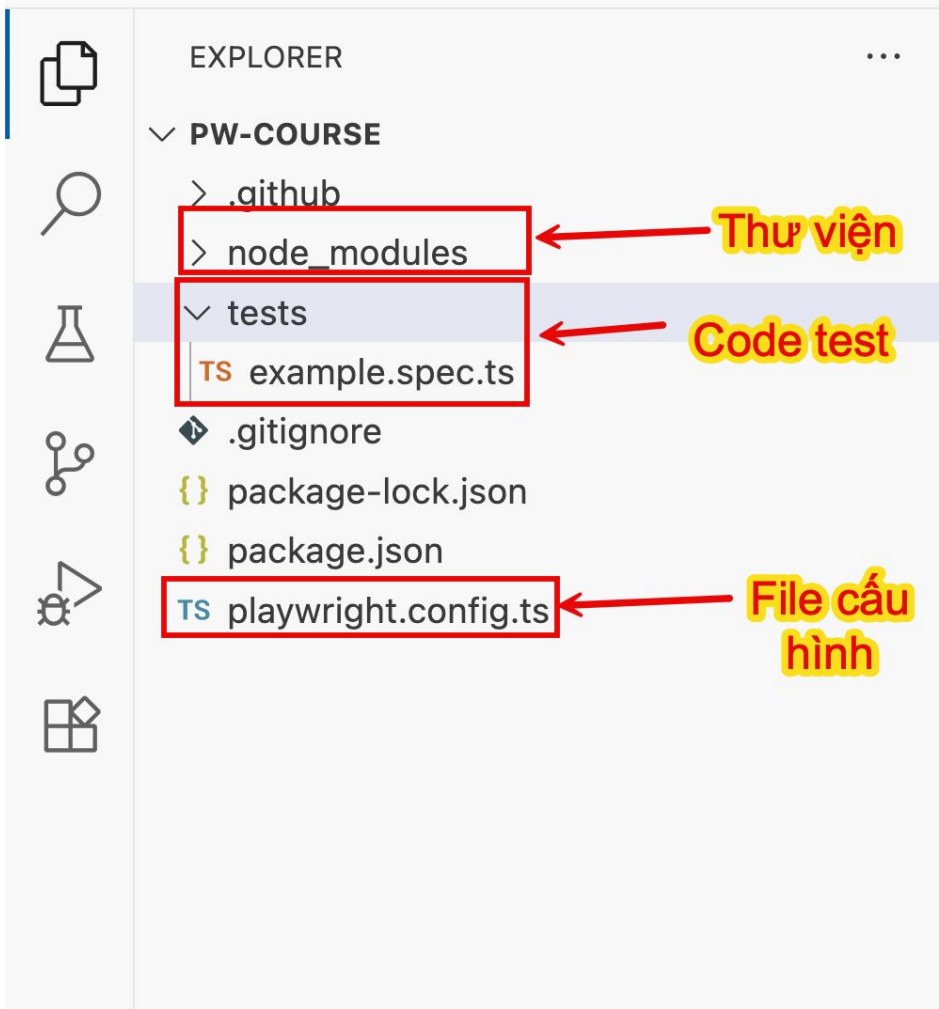
- ./tests/example.spec.ts - Example end-to-end test
- ./playwright.config.ts - Playwright Test configuration

Visit <https://playwright.dev/docs/intro> for more information. ✨

Happy hacking! 🤖

```
alex@m4 ~/w/t/p/pw-course>
```

**Thành công**



TS example.spec.ts X

tests &gt; TS example.spec.ts &gt; ...

```
1 import { test, expect } from '@playwright/test';
2
3 test('has title', async ({ page }) => {
4   await page.goto('https://playwright.dev/');
5
6   // Expect a title "to contain" a substring.
7   await expect(page).toHaveTitle(/Playwright/);
8 });
```

**Bấm vào để chạy test****Các test**

```
10 test('get started link', async ({ page }) => {
11   await page.goto('https://playwright.dev/');
12
13   // Click the get started link.
14   await page.getByRole('link', { name: 'Get started' }).click();
15
16   // Expects page to have a heading with the name of Installation.
17   await expect(page.getByRole('heading', { name: 'Installation' })).toBeVisible();
18 });
19
```

TESTING

TEST EXPLORER

Filter (e.g. text, !exclude, @tag)

No test results yet.

- tests
  - playwright.config.ts [firefox] — disabled
  - playwright.config.ts [webkit] — disabled

PLAYWRIGHT

TOOLS

- Pick locator
- Record new
- Record at cursor
- Reveal test output
- Close all browsers

PROJECTS

- ☒ chromium
- ☐ firefox
- ☐ webkit

SETUP

- Run global setup
- Run global teardown
- Clear cache

SETTINGS

- ☒ Show browser — one worker
- ☐ Show trace viewer

TS example.spec.ts

```
tests > TS example.spec.ts > ...
1 import { test, expect } from '@playwright/test';
2
3 test('has title', async ({ page }) => {
4   await page.goto('https://playwright.dev/');
5
6   // Expect a title "to contain" a substring.
7   await expect(page).toHaveTitle(/Playwright/);
8 });
```

Nếu không hiện nút run hoặc không hiện browser

1. Click vào tab test
2. Click "refresh" icon
3. Chọn "chromium" ở mục Projects
4. Chọn "Show browser" ở mục Settings

}).click  
f Installa  
stallation

# Hiểu “vẹt” Playwright code



Code Playwright rất dễ hiểu  
Tương tự như “lời thoại”

VD:

- goto = đi đến
- expect = mong muốn
- toHave... = có...

```
import { test, expect } from '@playwright/test';

test('has title', async ({ page }) => {
  await page.goto('https://playwright.dev/');

  // Expect a title "to contain" a substring.
  await expect(page).toHaveTitle(/Playwright/);
});
```

```
2
3 test('has title', async ({ page }) => {
4   await page.goto('https://playwright.dev/');
5
6   // Expect a title "to contain" a substring.
7   await expect(page).toHaveTitle(/Playwright/);
8 });
```

Đi đến trang playwright.dev

Kiểm tra tiêu đề có  
chứa chữ "Playwright" không

```
test('get started link', async ({ page }) => {  
  await page.goto('https://playwright.dev/');  
  
  // Click the get started link.  
  await page.getByRole('link', { name: 'Get started' }).click();  
  
  // Expects page to have a heading with the name of Installation.  
  await expect(page.getByRole('heading', { name: 'Installation' })).toBeVisible();  
});
```

← Đi đến trang playwright.dev

← Click vào link có chữ "Get started"

↑  
← Kiểm tra heading "Installation" có  
Hiển thị trên màn hình

# Hiểu “vẹt” Playwright code



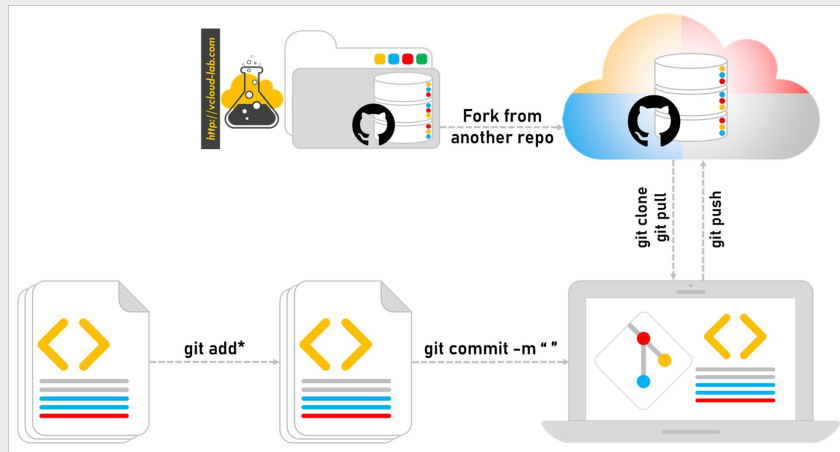
*Đừng lo, bạn sẽ sớm “hiểu bản chất” và “tư duy” được những dòng code này.*

Hãy tập trung xây dựng một **nền tảng vững chắc** về lập trình trong 4 buổi đầu nhé ^^



# Đưa code lên GitHub

git push origin main



# Đưa code lên GitHub: tạo repo

Khởi tạo repo local  
**git init**

Tạo repo GitHub và liên kết tới repo local  
**git remote add origin <url>**

Thêm file vào staging  
**git add .**

Commit file  
**git commit -m "<message>"**

Push code  
**git push origin main**

Làm **1 lần duy nhất**

Làm mỗi khi  
có thay đổi

# Đưa code lên GitHub: tạo repo

- Tạo repo
  - Truy cập:  
<https://github.com/new>
  - Điền tên repository
  - Chọn “Public”

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

Owner \* / Repository name \* 1  
minhphong306 / pw-course  
pw-course is available.

Great repository names are short and memorable. Need inspiration? How about curly-pancake ?

Description (optional)

Public 2  
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:

☐ Add a README file  
This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None

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

Choose a license

License: None

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

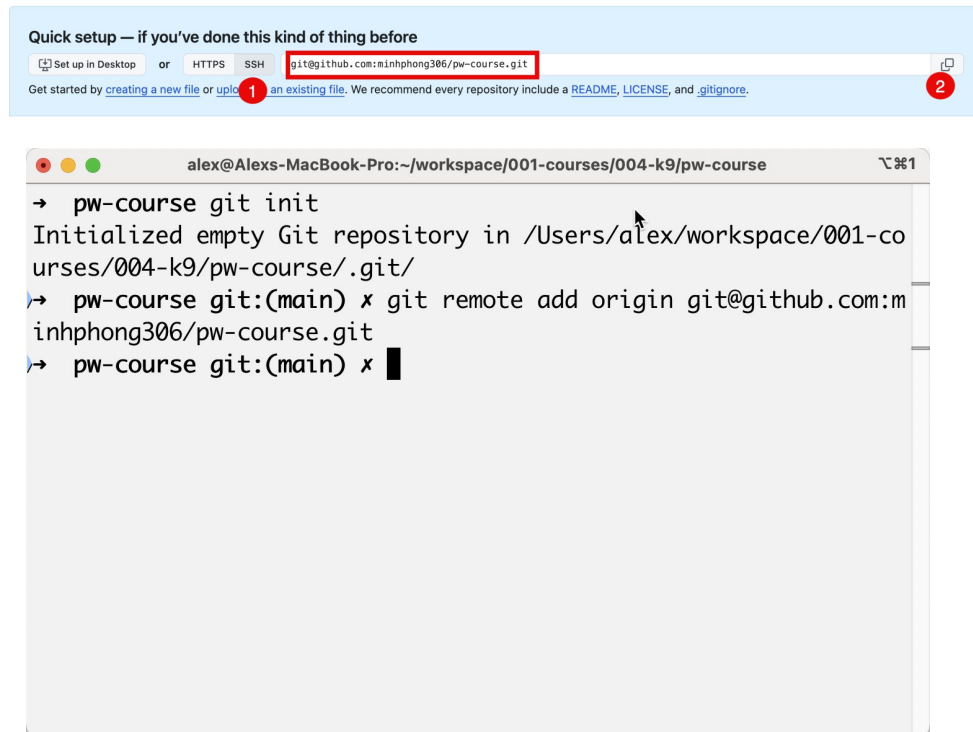
1 You are creating a public repository in your personal account.

Create repository

3

# Đưa code lên GitHub: khởi tạo

- Tạo repo
- Khởi tạo
  - Khởi tạo repo local: `git init`
  - Liên kết repository vừa tạo với Git: `git remote add origin <ssh_link>`
  - Thêm code: `git add .`
  - Thêm commit: `git commit -m"init project"`



The image shows two screenshots. The top one is the GitHub 'Quick setup' page, which has a light blue header. It says 'Quick setup — if you've done this kind of thing before'. Below this, there are three buttons: 'Set up in Desktop', 'HTTPS', and 'SSH'. The 'SSH' button is selected, and a text box next to it contains the SSH link 'git@github.com:minhphong306/pw-course.git', which is highlighted with a red box. Below the buttons, there is a line of text: 'Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).' There are two red circles with numbers '1' and '2' next to the links. The bottom screenshot is a terminal window with a title bar that says 'alex@Alexs-MacBook-Pro: ~/workspace/001-courses/004-k9/pw-course'. The terminal shows the following commands and output: 'pw-course git init' followed by 'Initialized empty Git repository in /Users/alex/workspace/001-courses/004-k9/pw-course/.git/'. Then, 'pw-course git:(main) x git remote add origin git@github.com:minhphong306/pw-course.git' and finally 'pw-course git:(main) x' followed by a cursor.

```
Quick setup — if you've done this kind of thing before
Set up in Desktop or HTTPS SSH git@github.com:minhphong306/pw-course.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

alex@Alexs-MacBook-Pro: ~/workspace/001-courses/004-k9/pw-course
→ pw-course git init
Initialized empty Git repository in /Users/alex/workspace/001-courses/004-k9/pw-course/.git/
→ pw-course git:(main) x git remote add origin git@github.com:minhphong306/pw-course.git
→ pw-course git:(main) x
```

# Đưa code lên GitHub: push code

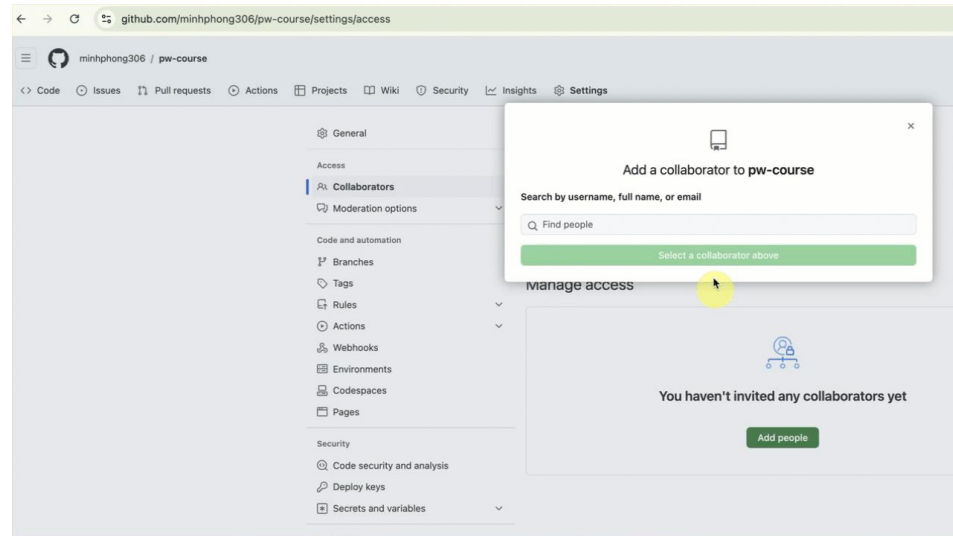
- Tạo repo
- Khởi tạo
- Push code
  - `git push origin main`

A terminal window with a title bar showing the user 'alex' on a 'MacBook-Pro' at the path '~/workspace/001-courses/004-k9/pw-course'. The terminal contains three lines of commands and their outputs: 1. 'pw-course git init' followed by 'Initialized empty Git repository in /Users/alex/workspace/001-courses/004-k9/pw-course/.git/'. 2. 'pw-course git:(main) x git remote add origin git@github.com:inhphong306/pw-course.git'. 3. 'pw-course git:(main) x' followed by a black cursor block.

```
alex@Alexs-MacBook-Pro:~/workspace/001-courses/004-k9/pw-course
→ pw-course git init
Initialized empty Git repository in /Users/alex/workspace/001-co
urses/004-k9/pw-course/.git/
→ pw-course git:(main) x git remote add origin git@github.com:m
inhphong306/pw-course.git
→ pw-course git:(main) x █
```

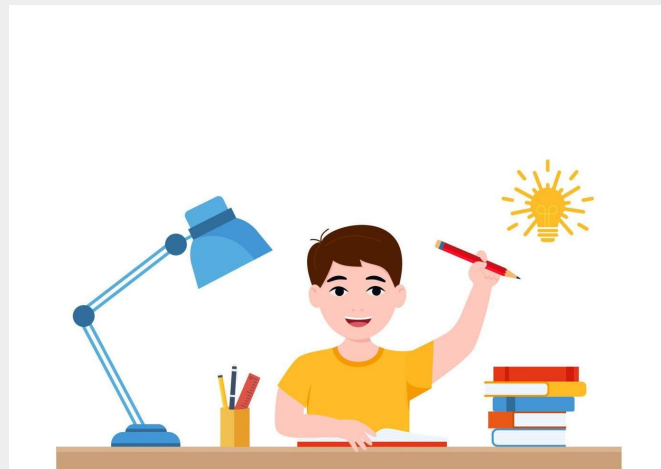
# GitHub: Invite collaborators

- Invite để mentor chấm bài
- Truy cập repo setting:  
**`https://github.com/<username>/<repo_name>/settings/access`**
- Add collaborator



# Bài tập về nhà

Practice makes perfect



## What's next

- Git & Javascript basic



# Thank you



[https://go.betterbytesvn.com/  
BBA-student-feedback](https://go.betterbytesvn.com/BBA-student-feedback)

Đánh giá chất lượng  
buổi học