

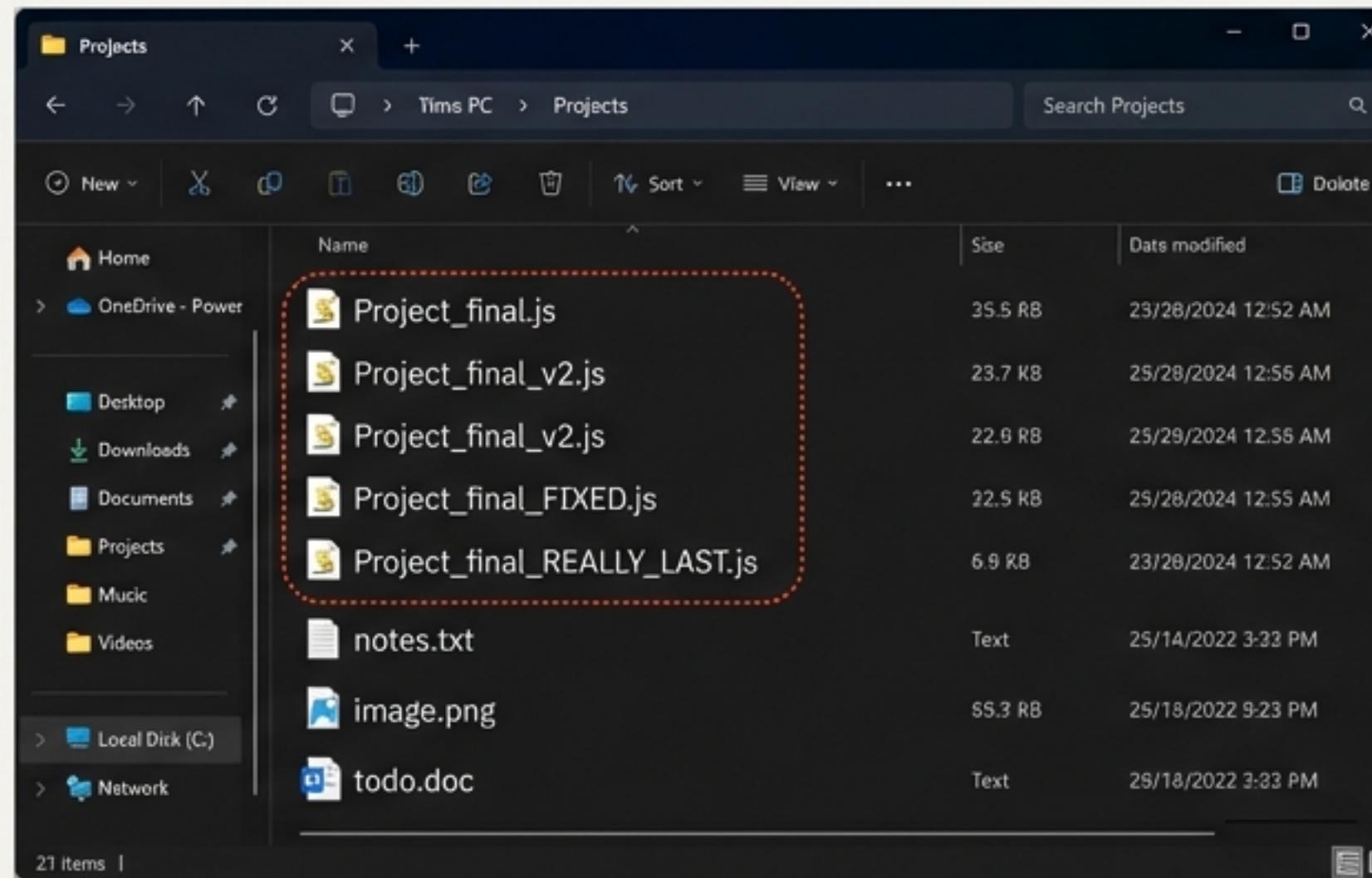


Git & GitHub 101: คู่มือฉบับนักเรียน (Step-by-Step)

เปลี่ยนการเขียนโค้ดให้เป็นระบบ หยุดปัญหาไฟล์หาย ย้อนเวลาแก้ไขได้ดั่งใจ

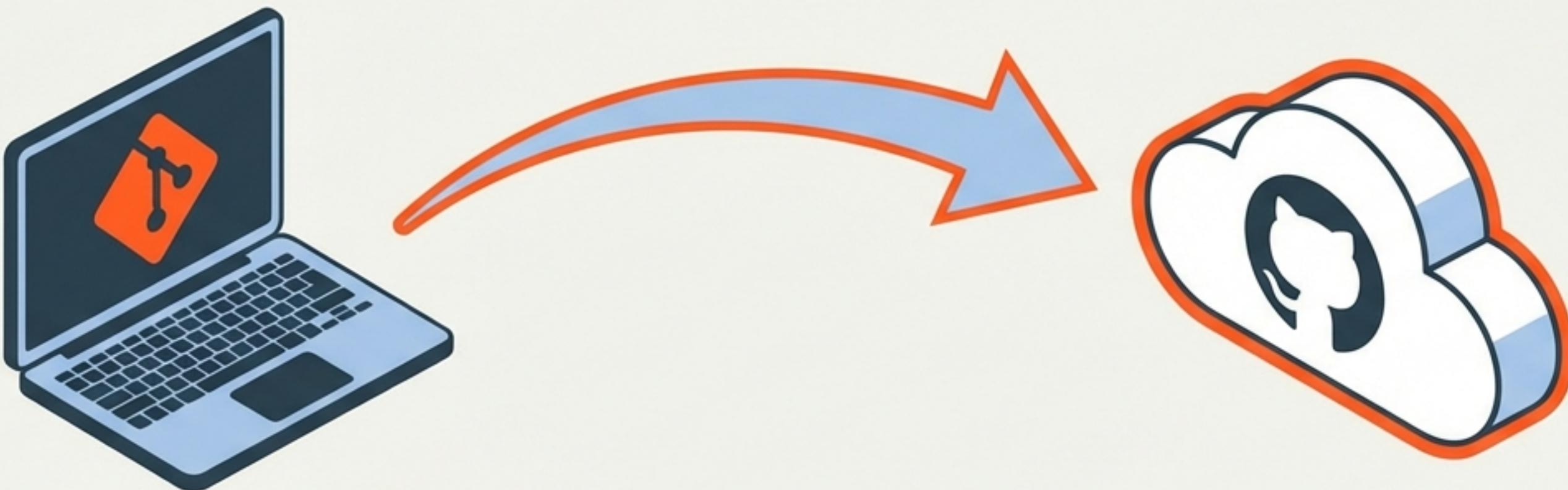
คุณเคยเจอปัญหาเหล่านี้ไหม?

- 1. **The Mess:** ไฟล์เยอะจนสับสน ไม่รู้อันไหนคือเวอร์ชันล่าสุดที่ใช้งานได้จริง
- 2. **The Fear:** เพลอลบໂຄດີດ หรือแก້ໂຄດແລ້ວພັງ ແຕ່ກູ້ອື່ນໄມ່ໄດ້ ຕ້ອງເຮີ່ມເບີຍໃໝ່
- 3. **The Struggle:** ทำงานกลຸມແລ້ວໄຟລໍກັບກັນ ຕ້ອງສ່າງงานຜ່ານ Flash Drive ຮີ້ວີ້ Google Drive ແລ້ວງວ່າຂອງໃກຣອັປເດຕສຸດ



ถึงเวลาบอกลาวีเดມๆ ແລ້ວมาຮູ້ຈັກກັບระบบ Version Control

Git และ GitHub ต่างกันอย่างไร?



Git (The Tool) = Time Machine

ระบบ **Version Control** ที่ติดตั้งในเครื่องของคุณ
กำหนดที่บันทึกทุกการเปลี่ยนแปลง (Save Points) ช่วยให้คุณ
‘ย้อนเวลา’ กลับไปดูได้ทุกเมื่อหากเกิดข้อผิดพลาด

GitHub (The Hub) = Social Network

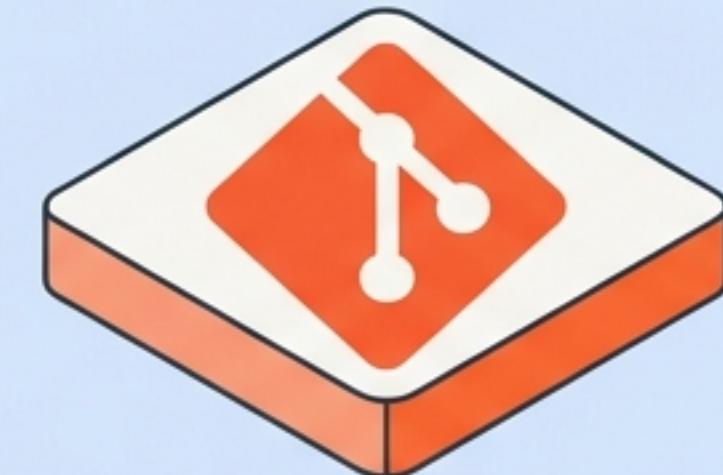
เว็บไซต์สำหรับฝากไฟล์ Git ไว้บน Cloud (Remote Server)
เป็นพื้นที่ส่วนกลางให้ทีมเข้ามาทำงานร่วมกัน
หรือใช้เป็น Portfolio โชว์ผลงาน

Git ทำงานแบบ Offline (ในเครื่อง) → GitHub ทำงานแบบ Online (บนเว็บ)

Step 0: เตรียมเครื่องมือให้พร้อม (Installation)



1. **สมัครบัญชี GitHub:** ไปที่ `github.com` (ใช้อีเมลจริงหรืออีเมลโรงเรียนเพื่อรับสิทธิพิเศษ)



2. **ดาวน์โหลด Git:** ไปที่ `git-scm.com` เลือก OS ของคุณ (Windows/Mac) แล้วกด Install (กด Next ตามค่า Default ได้เลย)



3. **เตรียม Editor:** แนะนำ **VS Code** เพื่อความสะดวกในการเขียนโค้ด

```
$ git --version  
git version 2.39.0
```

Pro Tip: เช็คว่าติดตั้งสำเร็จไหม? เปิด Terminal (Mac) หรือ Command Prompt (Windows) และพิมพ์คำสั่ง: `git --version` (ถ้ามีเลขเวอร์ชันขึ้นแสดงว่าพร้อมลุย!)

Step 1: ยืนยันตัวตน (Identity Setup)



ก่อนเริ่มบันทึกงาน ต้องบอก Git ก่อนว่า ‘ใคร’ เป็นคนเขียนโค้ดนี้ (ทำครั้งเดียวหลังติดตั้ง) เพื่อให้เพื่อนร่วมทีมรู้ว่าใครเป็นคนแก้ไขไฟล์

```
git config --global user.name "ชื่อของคุณ  
(ภาษาอังกฤษ)"  
git config --global user.email "อีเมลของคุณ  
(ที่ใช้สมัคร GitHub)"
```

ตรวจสอบค่าที่ตั้งไว้ด้วยคำสั่ง `git config --list`

Step 2: สร้างพื้นที่ทำงาน (Initialize)



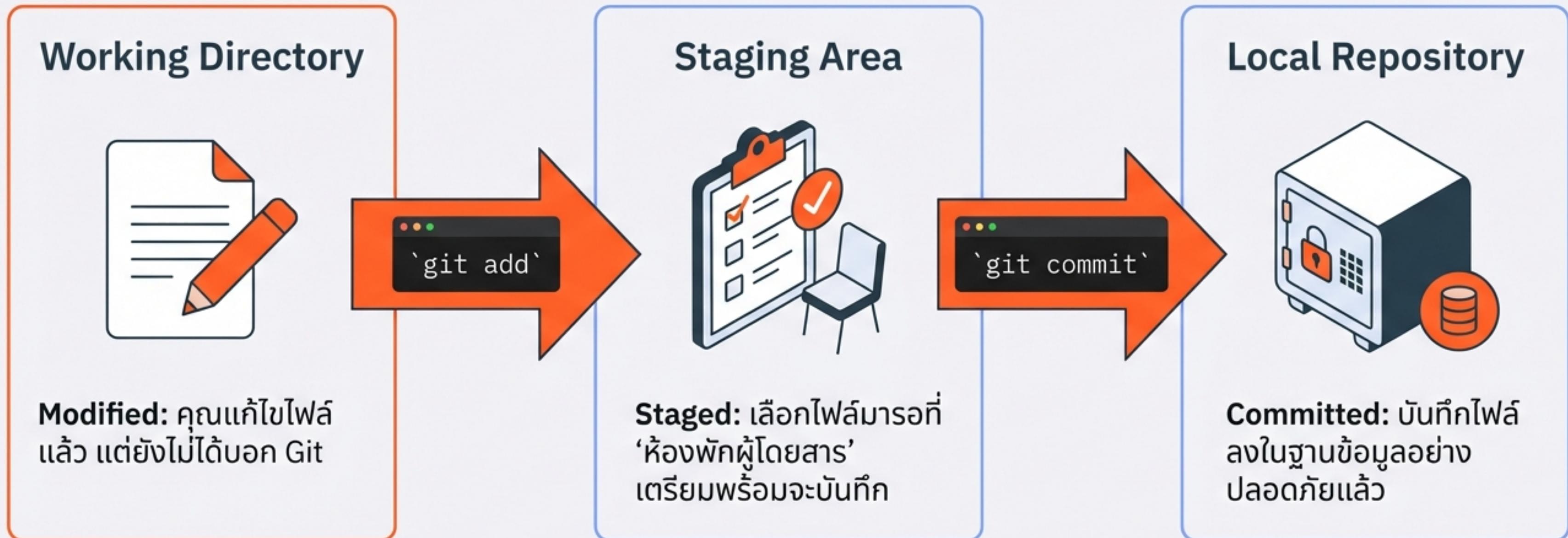
เปลี่ยนโฟลเดอร์โปรเจกต์ธรรมดา
ให้กลายเป็น **Local Repository**
(ที่มีระบบติดตามไฟล์)

1. เปิด Terminal แล้ว `cd` เข้าไปที่โฟลเดอร์โปรเจกต์ของคุณ
2. พิมพ์คำสั่ง:

```
git init
```

Result: คุณจะได้โฟลเดอร์ลับ ` .git ` (Hidden Folder) ปรากฏขึ้นมา
นี่คือ ‘กล่องดำ’ ที่ใช้เก็บประวัติการแก้ไขทั้งหมด ห้ามลบเด็ดขาด!

เข้าใจวงจรชีวิตของไฟล์ (The Git Lifecycle)



Step 3: เลือกไฟล์ที่จะบันทึก (Staging)



Git จะไม่บันทึกทุกไฟล์อัตโนมัติ เราต้อง “เลือก” เวลาไฟล์ที่ต้องการก่อน (เหมือนเลือกของใส่ตะกร้าก่อนจ่ายเงิน)

เลือกทีละไฟล์:

```
git add index.html
```

****เลือกทุกไฟล์ (นิยมใช้)**:**

```
git add .
```

จุดหมายถึง “ทั้งหมดในโฟลเดอร์นี้”

Status Check: ใช้คำสั่ง `git status` เพื่อดูว่าไฟล์ไหนเป็นสีแดง (ยังไม่เลือก) หรือสีเขียว (พร้อมบันทึกแล้ว)

Step 4: บันทึกความตรงจ้า (Commit)



ยืนยันการเปลี่ยนแปลงลงในฐานข้อมูล เปรียบเสมือนการสร้าง Checkpoint ที่เราสามารถย้อนกลับมาได้เสมอ

```
git commit -m "อธิบายสิ่งที่แก้ไข เช่น เพิ่มหน้า Login หรือ แก้ไขสีปุ่ม"
```

Best Practice

ข้อความใน `\"` (Log Message) สำคัญมาก! ห้ามพิมพ์มั่ว ควรเขียนให้รู้เรื่องว่าเวอร์ชันนี้ทำอะไรไปเพื่อให้ตัวคุณเองในอนาคตกลับมาอ่านแล้วเข้าใจ

Step 5: สร้างบ้านบน Cloud (Create Remote Repo)

1. Login เข้า `github.com`
2. คลิกปุ่ม (+) บนขวาบน เลือก '**New repository**'
3. **Repository name:** ตั้งชื่อโปรเจกต์ (ภาษาอังกฤษ)
4. **Visibility:** เลือก Public (ทุกคนเห็น)
หรือ Private (ส่วนตัว)
5. กด **Create repository**



Step 6: เชื่อมต่อและอัปโหลด (Connect & Push)



นำโค้ดจากเครื่องเรา (Local)
ขึ้นไปเก็บไว้บน GitHub (Remote)

1. เชื่อมต่อ (ทำครั้งแรก)

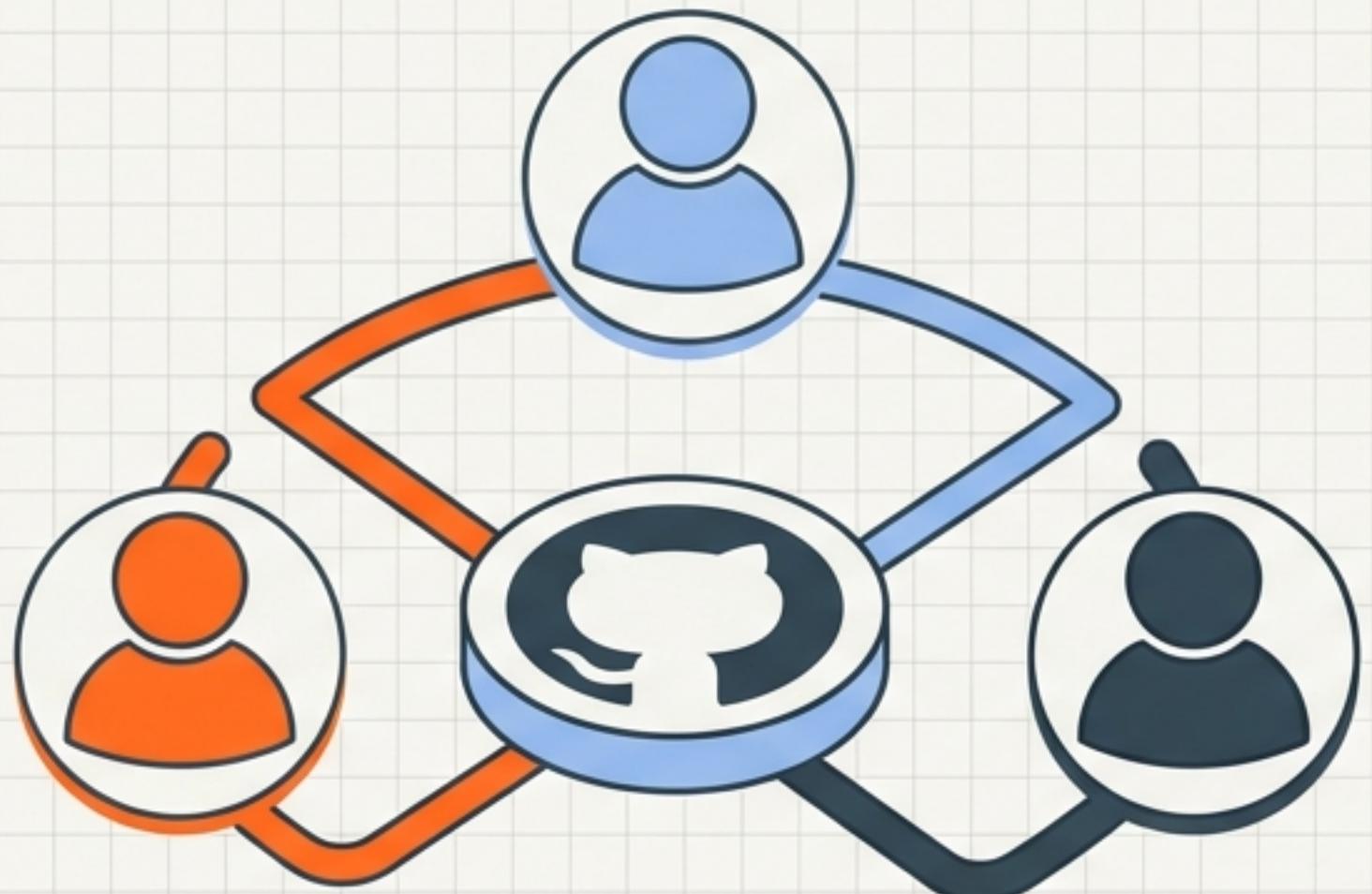
```
git remote add origin  
https://github.com/ชื่อคุณ/ชื่อโปรเจกต์.git
```

2. อัปโหลด (Push)

```
git push -u origin main
```

‘origin’ คือชื่อเล่นของ Server ปลายทาง, ‘main’ คือชื่อสาขา (Branch) หลักที่เราทำงานอยู่

เมื่อต้องย้ายเครื่องหรือทำงานกลุ่ม



Git Clone

เมื่อต้องการก็อปปี้โปรเจกต์จาก GitHub มาลงเครื่องใหม่
(เช่น เปลี่ยนจากคอมฯ โรงเรียนมาทำที่บ้าน)



```
git clone <url_ของ_repo>
```

Git Pull

เมื่อต้องการดึงอัปเดตล่าสุดที่เพื่อนทำไว้
(หรือเราทำไว้เครื่องอื่น) มาใส่เครื่องปัจจุบัน



```
git pull
```

*Rule of Thumb**: ‘Pull ก่อนเริ่มงานเสมอ’ เพื่อให้โค้ดของเราเป็นปัจจุบันที่สุด

Special Bonus: GitHub Student Developer Pack



สิทธิพิเศษสำหรับนักเรียน/นักศึกษา!
รับเครื่องมือระดับ Pro มูลค่าหลายหมื่นบาทฟรี

- **What you get :** GitHub Copilot ฟรี,
Hosting/Domain ฟรี, JetBrains IDEs
ครบชุด
- **Requirements :** อายุ 13 ปีขึ้นไป,
มีอีเมลโรงเรียน (.edu) หรือบัตรนักเรียน

สมัครเลยที่ [`education.github.com/pack`](https://education.github.com/pack)



Git Command Cheat Sheet (จดไว้ใช้สอบ!)

git init

เริ่มต้นระบบ Git

git status

เช็คสถานะไฟล์

git add .

เตรียมบันทึกทุกไฟล์

git commit -m "..."

บันทึกเวอร์ชัน (Save)

git push

ส่งงานขึ้น GitHub

git pull

ดึงงานจาก GitHub

git log

ดูประวัติการแก้ไขย้อนหลัง

ไปต่อให้สุด! แหล่งศึกษาเพิ่มเติม

YouTube: KongRuksiam Git



ค้นหา 'KongRuksiam Git' สำหรับคลิปสอนภาษาไทยที่ละเอียดและเข้าใจง่าย

Official Docs



`docs.github.com`
สำหรับข้อมูลเชิงลึก

“การเขียนโค้ดคือการเดินทาง Git คือเข็มทิศที่จะช่วยให้คุณไม่หลงทาง เริ่มต้นสร้างโปรเจกต์แรกของคุณวันนี้!”

