

✓ วิธีแก้ไข

คุณต้องใส่ข้อความบรรยายการ commit ไว้ในเครื่องหมายคำพูด เช่น:

```
hash
1 git commit -m "เริ่มต้นโปรเจกต์แปลนิยาย"
```

หรือ

```
hash
1 git commit -m "เพิ่ม glossary พื้นฐานและ trans1.py"
```

🔧 ตัวอย่างที่ถูกต้อง

```
hash
1 (venv) C:\Users\ZE HOUSE\Trans1> git commit -m "เริ่มต้นโปรเจกต์แปลนิยาย"
2 [master (root-commit) 9f3a4b1] เริ่มต้นโปรเจกต์แปลนิยาย
3 2 files changed, 500+ insertions(+)
4 create mode 100644 glossary.yaml
5 create mode 100644 trans1.py
```

✓ แบบนี้คือสำเร็จ!

✖ เคล็ดลับการเขียน Commit Message ให้ดี

สั้น กระชับ อ่านเข้าใจ	"เพิ่มระบบ reprocess"
เริ่มต้นด้วยกริยา	"เพิ่ม", "แก้ไข", "อัปเดต", "ลบ"
ระบุเป้าหมาย	"เปลี่ยน 'จางซาน' → 'จางเชียนเซิง'"
ใช้ภาษาไทยก็ได้	Git รองรับ Unicode สมบูรณ์

ตัวอย่างข้อความดี ๆ:

- git commit -m "เพิ่ม trans1.py และโครงสร้างโฟลเดอร์"
- git commit -m "แก้ไข glossary: ปรับชื่อสำนักหอกฟ้า → ฉลองเทียน"
- git commit -m "เพิ่มระบบ diff_report.html"

✓ ตรวจสอบประวัติการ commit

หลังจาก `commit` สำเร็จ คุณสามารถดูประวัติได้ด้วย:

```
hash
1 git log --oneline
```

ผลลัพธ์จะประมาณนี้:

```
hash
1 9f3a4b1 (HEAD -> master) เริ่มต้นโปรเจกต์แปลนิยาย
```

✓ สรุปคำสั่งที่ควรใช้ต่อไป

```
hash
1 # หลังแก้ glossary.yaml
2 git add glossary.yaml
3 git commit -m "เปลี่ยนชื่อตัวละครตาม feedback"
4
5 # รัน reprocess
6 python reprocess_all_outdated.py
7
8 # ถ้ามีไฟล์ใหม่เพิ่ม
9 git add .
10 git commit -m "เพิ่มรายงาน diff และระบบ cleanup"
```

✓ ขั้นตอนต่อไปที่แนะนำ

1. แก้ <code>glossary.yaml</code>	เปิดด้วย VS Code แล้วแก้ไข
2. บันทึกไฟล์	Ctrl + S
3. เพิ่มลง Git	<code>git add glossary.yaml</code>
4. Commit	<code>git commit -m "อัปเดตศัพท์บทที่ 1-5"</code>
5. อัปเดตบทแปล	<code>python reprocess_all_outdated.py</code>