ใบงาน การใช้งาน GitHub ในการทำงานจากหลายสถานที่

ขั้นตอนที่ 1 เปิด Terminal หรือ Command Prompt

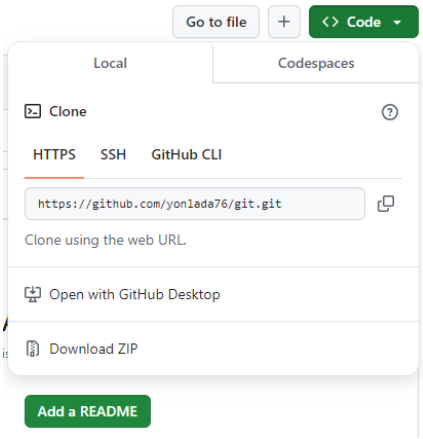
ตรงจุดนี้นักศึกษาสามารถใช้Git Bash Terminal บน vscode หรือ ใช้Git Bash หรือ Command Prompt ก็ได้

-ใช้ Git Bash Terminal บน vscode

ขั้นตอนที่ 2 Clone repository ลงในเครื่อง

1. เข้าไปที่หน้า GitHub repository ของตนเอง

2. คลิกที่ปุ่ม Code (ปุ่มสีเขียว) แล้วคัดลอก URL ของ repository (เลือก HTTPS หรือ SSH)



3. เปิด Terminal หรือ Command Prompt แล้วใช้ค าสั่ง git clone เพื่อคัดลอก repository มายังเครื่องใหม่

-git clone https://github.com/yonlada76/git.git

ขั้นตอนที่ 3 เข้าไปในโฟลเดอร์ของ repository

1. ใช้คำสั่ง cd เพื่อเข้าไปในโฟลเดอร์ของ repository

-cd repository-name คือ cd 'c:/Users/COM/Desktop/git1/git'

```
COM@904-40 MINGW64 ~/Desktop/git1
$ cd 'c:/Users/COM/Desktop/git1/git'

COM@904-40 MINGW64 ~/Desktop/git1/git (main)
$
```

ขั้นตอนที่ 4ใช้คำสั่ง git pull

1. ใช้คำสั่ง git pull เพื่อดึงข้อมูลล่าสุดจาก GitHub

-git pull origin master ต้องใช้คำสั่ง git pull origin main เพราะเป็นชื่อของ ranch ที่ต้องการดึงการเปลี่ยนแปลง โดย `main` มักจะเป็น branch หลักใน repository ที่ใช้แทน `master` ใน Git

```
COM@904-40 MINGW64 ~/Desktop/git1/git (main)
$ git pull origin main
From https://github.com/yonlada76/git
 * branch            main       -> FETCH_HEAD
Already up to date.

COM@904-40 MINGW64 ~/Desktop/git1/git (main)
$
```

ขั้นตอนที่ 5 ดูประวัติการ Commit

1. ใช้คำสั่ง git log เพื่อดูประวัติการ commit
   - commit hash ชุดตัวอักษรและตัวเลขที่ไม่ซ้ ากัน (ใช้ในการอ้างอิง commit นั้น)
   - ชื่อผู้ที่ท า commit
   - วันที่ commit
   - ข้อความ commit ที่บอกว่าได้ท าการเปลี่ยนแปลงอะไร

-git log การเรียกดูประวัติของ commit ว่ามีงานใน git ที่เราใส่ลิงก์ไปว่ามีงานเท่าไรบ้าง

```
COM@904-40 MINGW64 ~/Desktop/git1/git (main)
$ git log
commit 890c8275572d2ce236c61e653915bcec668f7ff1 (HEAD -> main, origin/main,
origin/HEAD)
Author: yonlada76 <655021000097@mail.rmutk.ac.th>
Date:   Tue Oct 8 12:27:11 2024 +0700

    Add files via upload

COM@904-40 MINGW64 ~/Desktop/git1/git (main)
$
```

2. คัดลอก commit hash ของ commit ที่คุณต้องการเรียกคืนหรือตรวจสอบขั้นตอนที่ 2 การเรียกคืนไปยัง commit ก่อนหน้า

-git reset --hard <commit-hash> ย้อนกลับไปยัง commit ก่อนหน้าโดยใช้git reset

```
COM@904-40 MINGW64 ~/Desktop/git1/git (main)
$ git reset

COM@904-40 MINGW64 ~/Desktop/git1/git (main)
$ git reset --hard 890c8275572d2ce236c61e653915bcec668f7ff1
HEAD is now at 890c827 Add files via upload

COM@904-40 MINGW64 ~/Desktop/git1/git (main)
$
```

- git revert <commit-hash> ยกเลิก commit โดยไม่ลบประวัติ(ใช้git revert)

```
COM@904-40 MINGW64 ~/Desktop/git1/git (main)
$
Revert "Add files via upload"

This reverts commit 890c8275572d2ce236c61e653915bcec668f7ff1.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#       deleted:    git1/index.html
#
~
~
~
~
~
~
~
~
~
~
```

คำถามท้ายใบงาน

1. อธิบายความแตกต่างระหว่างคำสั่ง git clone และ git pull

- **git clone**: ใช้สำหรับคัดลอก repository ทั้งหมดมาครั้งแรก

git pull: ใช้เพื่อดึงการเปลี่ยนแปลงใหม่ๆ ที่เกิดขึ้นใน repository มายัง local เมื่อคุณมี repository อยู่แล้ว

2. การใช้คำสั่ง git revert และ git reset มีความแตกต่างกันอย่างไร?

- git revert สร้าง commit ใหม่ที่ทำการ "ย้อนกลับ" การเปลี่ยนแปลงของ commit ที่กำหนด ทำให้ประวัติยังคงอยู่และปลอดภัยสำหรับการแชร์บน remote repository

git reset ย้ายตำแหน่ง HEAD และสามารถลบ commit ได้ ทำให้สามารถลบประวัติการเปลี่ยนแปลง และไม่ควรใช้เมื่อมีการแชร์ commit นั้นไปยัง remote repository แล้ว

3. หากคุณทำงานร่วมกันในทีมและต้องการแก้ไขโค้ดที่ส่งผลกระทบกับหลายคน คุณควรเลือกใช้คำสั่งใดระหว่าง git reset หรือ git revert?

-ใช้ **git revert** เพื่อย้อนกลับการเปลี่ยนแปลง เพราะเป็นวิธีที่ปลอดภัยและมีประวัติการเปลี่ยนแปลงที่ชัดเจน ช่วยป้องกันปัญหาความขัดแย้งที่อาจเกิดขึ้นจากการลบ commit ด้วยคำสั่ง git reset