**LabGitWorkshop**

**คำถาม**

1. **Git เก็บข้อมูลไว้ที่ไหน?**  
   - บน Github เป็น remote repository  
   - บนเครื่อง (PC, Mac, Notebook) เป็น local repository  
   - โฟลเดอร์ “.git” เก็บข้อมูลที่เกี่ยวข้องกับระบบข้อมูล
2. **อธิบายวิธีการบันทึกการเปลี่ยนแปลงใน Git ?**- ในการบันทึกการเปลี่ยนแปลงไฟล์ต่างๆ ใน Git จะให้คำสั่ง “ git commit ”
3. **อธิบายวิธิการบันทึกเกี่ยวกับการเปลี่ยนแปลง ทำอย่างไร?**- ในการบันทึกการเปลี่ยนแปลงไฟล์ต่างๆ ใน Git มีขั้นตอนวิธีการ  
   ดังนี้  
    1. ใช้คำสั่ง “ git status ” เพื่อเช็คว่ามีไฟล์ใดบ้าง ที่มีการแก้ไขเกิดขึ้น  
    2. ใช้คำสั่ง “ git add <ชื่อไฟล์> ” เพื่อเพิ่มไฟล์ที่ต้องการจะบันทึก (เลือกไฟล์ที่ พึ่งแก้ไขไป)  
    3. ใช้คำสั่ง “ git commit -m ‘คำอธิบายเกี่ยวกับการเปลี่ยนแปลง’ ” เพื่อบันทึก การเปลี่ยนแปลงใน repository

**วัตถุประสงค์**

1. **สร้างที่เก็บ Git ในเครื่อง**
2. **ดำเนินการแก้ไข-เพิ่ม-คอมมิตสำหรับไฟล์ตั้งแต่หนึ่งไฟล์ขึ้นไป**

**Remotes in GitHub**

* **นักศึกษา สมัคร** [**https://github.com/**](https://github.com/) **✓**
* **จากนั้น repositories, เลือก New repository ✓**
* **สร้าง repository ชื่อ LabCSD2301 ✓**
* **A screenshot of a computer

  Description automatically generatedผลลัพธ์ ✓**

**Setting Up Git**

**แสดงผลลัพธ์ที่ได้จากการทำงานของคำสั่งที่กำหนด และอธิบาย**

**Git bash**

* **git config –global user.name “your username”**

**  
- เซตชื่อผู้ใช้**

* **git config –global user.email “your email”  
    
  - เซตอีเมล**
* **git config –global color.ui “auto”  
  **

**- เซต color.ui ให้เป็นแบบ auto**

1. **Create git**

**#ทำการสร้าง folder ชื่อ TheDataShop**

**mkdir TheDataShop**

****

**#access เข้าไป TheDataShop directory**

**cd TheDataShop**

**A black screen with white text

Description automatically generated**

**#check that you are in TheDataShop (เช็คว่าอยู่ในโฟลเดอร์ TheDataShop แล้ว)**

**pwd**

**A black background with white text

Description automatically generated**

**#check TheDataShop contents (list files) (เช็คว่าโฟลเดอร์ TheDataShop มีไฟล์อะไรบ้าง)**

**ls**

****

**#To show hidden files (เช็คว่ามีไฟล์ที่ซ่อนอยู่หรือไม่)**

**A black background with purple and blue text

Description automatically generated**

**ls -a**

**จากนั้น**

1. **echo “Day 1 notes” > notes.txt  
   A black background with colorful text

   Description automatically generated  
   สร้างไฟล์ notes.txt และ พิมพ์ข้อความในไฟล์นั้นว่า “Day 1 notes”**
2. **ls  
   A black background with text and numbers

   Description automatically generated  
   เช็คไฟล์ในโฟลเดอร์ พบว่ามีไฟล์ notes.txt**
3. **cat note.txt  
   A black screen with white text

   Description automatically generated  
   อ่านไฟล์ notes.txt มีข้อความอยู่ข้างใน**
4. **git status  
   A black background with white text

   Description automatically generated  
   เช็คสถานะของgit ไม่เจอ git repository (โฟลเดอร์ .git)**

**จากนั้น**

1. **git init  
   A screen shot of a computer

   Description automatically generated  
   สร้าง git repository (โฟลเดอร์ .git) ในโฟลเดอร์ TheDataShop**
2. **ls  
   A black background with text and numbers

   Description automatically generated  
   เช็คไฟล์ในโฟลเดอร์ พบว่ามีไฟล์ notes.txt**
3. **ls -a  
   A screen shot of a computer

   Description automatically generated  
   เช็คไฟล์ที่ซ่อน เจอโฟลเดอร์ .git และไฟล์ notes.txt**
4. **git status  
   A screen shot of a computer

   Description automatically generated  
   เช็คสถานะของ git**
5. **git add notes.txt   
     
   เพิ่ม notes.txt เข้าไปใน git repository**
6. **git status  
   A screen shot of a computer

   Description automatically generated  
   เช็คสถานะของ git**
7. **git commit -m “Start notes for data workshop”  
   A screen shot of a computer code

   Description automatically generated  
   บันทึกการเปลี่ยนแปลงใน git repository โดยกำหนดคำอธิบายของการเปลี่ยนแปลงว่า “Start notes for data workshop”**
8. **git status  
   A black screen with white text

   Description automatically generated  
   เช็คสถานะของ git**
9. **git log  
   A screen shot of a computer

   Description automatically generated  
   เก็บประวัติการเปลี่ยนแปลงของ branch**
10. **git log –oneline  
    A black screen with yellow text

    Description automatically generated  
    เก็บประวัติการเปลี่ยนแปลงของ branch แบบบรรทัดเดียว**