

### คำสั่ง git log

คำสั่ง git log ใช้ในการดูประวัติของ commit ใน repository โดยสามารถเพิ่มตัวเลือก (parameters) เพื่อกรองและจัดรูปแบบการแสดงผลได้

Parameters ที่เป็นไปได้:

1. **--oneline**: แสดง commit ID และข้อความ commit แบบย่อในแต่ละบรรทัด
2. **--graph**: แสดงกราฟของ commit history
3. **--decorate**: แสดงข้อมูลเพิ่มเติมเกี่ยวกับ branch, tag, และ references อื่นๆ
4. **--stat**: แสดงไฟล์ที่ถูกแก้ไขและสถิติเกี่ยวกับการเปลี่ยนแปลงในแต่ละ commit
5. **--pretty=format:"%h - %an, %ar : %s"**: กำหนดรูปแบบการแสดงผล commit log
6. **--merges**: แสดงเฉพาะ commit ที่เป็นการ merge
7. **-p** หรือ **--patch**: แสดงรายละเอียดการเปลี่ยนแปลงของแต่ละ commit
8. **-n <number>**: แสดงจำนวน commit ล่าสุดตามที่ระบุ
9. **--since="date"** และ **--until="date"**: แสดง commit ที่เกิดขึ้นในช่วงเวลาที่กำหนด
10. **--author=<name>**: แสดง commit ที่ถูกสร้างโดย author ตามที่ระบุ
11. **--grep=<pattern>**: แสดง commit ที่ข้อความ commit ตรงกับ pattern ที่ระบุ

คำสั่งที่ได้ใช้:

- git log --merges: ใช้เพื่อแสดงเฉพาะ commit ที่เป็นการ merge pull request
- git log --oneline: ใช้ในการดู commit log แบบย่อ
- git log -p: แสดงรายละเอียดการเปลี่ยนแปลงใน commit

- 
- git log --merges: ใช้ในการตอบคำถามเกี่ยวกับ pull requests
  - git log --oneline: ใช้เพื่อแสดง commit log แบบย่อ
  - git log -p: ใช้เพื่อแสดงรายละเอียดการเปลี่ยนแปลงของไฟล์
  - git log -1 --format=%cd <commit-id>: ใช้ในการดูวันที่ของ commit แรก

### คำสั่ง git show

คำสั่ง git show ใช้เพื่อแสดงข้อมูลเกี่ยวกับ commit, tag, หรือ object อื่นๆ ใน repository โดยจะให้ข้อมูลที่ละเอียดกว่า git log

Parameters ที่เป็นไปได้:

1. **<commit>**: แสดงรายละเอียดของ commit ที่ระบุ เช่น commit message, author, วันที่, และการเปลี่ยนแปลงของไฟล์
2. **--stat**: แสดงสถิติการเปลี่ยนแปลงในแต่ละไฟล์
3. **--name-only**: แสดงเฉพาะชื่อไฟล์ที่ถูกแก้ไข
4. **--name-status**: แสดงสถานะของไฟล์ (เช่น Modified, Added, Deleted) พร้อมกับชื่อไฟล์
5. **--pretty=format:"%h - %an, %ar : %s"**: กำหนดรูปแบบการแสดงผลของข้อมูล commit
6. **--patch** หรือ **-p**: แสดงการเปลี่ยนแปลงของไฟล์ที่ถูกแก้ไขใน commit
7. **<branch>**: ใช้ดูข้อมูลการเปลี่ยนแปลงของ branch ที่ระบุ

คำสั่งที่ได้ใช้:

- git show <commit-id>: ใช้เพื่อดูรายละเอียด commit ที่ระบุ
- git show --name-only <commit-id>: ใช้เพื่อแสดงเฉพาะไฟล์ที่ถูกแก้ไขใน commit นั้น

- 
- git show <commit-id>: ใช้เพื่อดูรายละเอียดของ commit
  - git show --name-only <commit-id>: ใช้ในการดูเฉพาะไฟล์ชื่อที่เปลี่ยนแปลง

### คำสั่ง git branch

คำสั่ง git branch ใช้เพื่อจัดการและตรวจสอบ branch ใน repository

Parameters ที่เป็นไปได้:

1. **-r**: แสดง branch ทั้งหมดที่อยู่ใน remote
2. **-a**: แสดง branch ทั้งหมดทั้ง local และ remote
3. **-v**: แสดง commit ล่าสุดของแต่ละ branch
4. **-vv**: แสดง commit ล่าสุดพร้อมการติดตาม branch (tracking branch)
5. **-d <branch>**: ลบ branch ที่ระบุ
6. **-D <branch>**: บังคับลบ branch ที่ระบุแม้ว่าจะยังไม่ได้ merge
7. **--merged <commit>**: แสดง branch ที่ถูก merge เข้ากับ commit ที่ระบุแล้ว
8. **--no-merged <commit>**: แสดง branch ที่ยังไม่ได้ merge เข้ากับ commit ที่ระบุ

คำสั่งที่ได้ใช้:

- git branch -a: ใช้เพื่อแสดง branch ทั้งหมดทั้งใน local และ remote
- git branch: ใช้เพื่อแสดง branch ที่อยู่ใน local repository

- 
- git branch -a: ใช้ในการดู branch ทั้งหมดใน local และ remote
  - git branch -r: ใช้ดูเฉพาะ remote branches
  - git branch: ใช้ดู branch ใน local

### คำสั่ง git diff

คำสั่ง git diff ใช้ในการตรวจสอบความแตกต่างระหว่างสองสถานะใน repository เช่นระหว่าง commit, branch, หรือไฟล์ใน staging และ working directory

Parameters ที่เป็นไปได้:

1. **<commit>**: แสดงความแตกต่างระหว่าง commit ปัจจุบันและ commit ที่ระบุ
2. **<commit>..<commit>**: แสดงความแตกต่างระหว่าง commit ที่ระบุสองตัว
3. **--name-only**: แสดงเฉพาะชื่อไฟล์ที่มีการเปลี่ยนแปลง
4. **--name-status**: แสดงชื่อไฟล์พร้อมสถานะการเปลี่ยนแปลง (เช่น Modified, Added, Deleted)
5. **--stat**: แสดงสถิติของการเปลี่ยนแปลงในแต่ละไฟล์
6. **\*\* -p หรือ --patch**: แสดงรายละเอียดของการเปลี่ยนแปลง
7. **--cached**: แสดงความแตกต่างระหว่าง staging area และ commit ล่าสุด
8. **--color**: ใช้สีในการแสดงผลความแตกต่าง
9. **<branch>**: แสดงความแตกต่างระหว่าง branch ปัจจุบันและ branch ที่ระบุ

คำสั่งที่ได้ใช้:

- git diff --name-only <tag1> <tag2>: ใช้เพื่อแสดงไฟล์ที่มีการเปลี่ยนแปลงระหว่างสอง tags
- git diff --stat: แสดงสถิติของไฟล์ที่ถูกแก้ไข

- 
- git diff --name-only <tag1> <tag2>: ใช้ในการดูไฟล์ที่เปลี่ยนแปลงระหว่างสอง tags
  - git diff --stat <commit-id>: ใช้ดูสถิติการเปลี่ยนแปลงของ commit