

Version-control

ทำไมต้องใช้ Version control

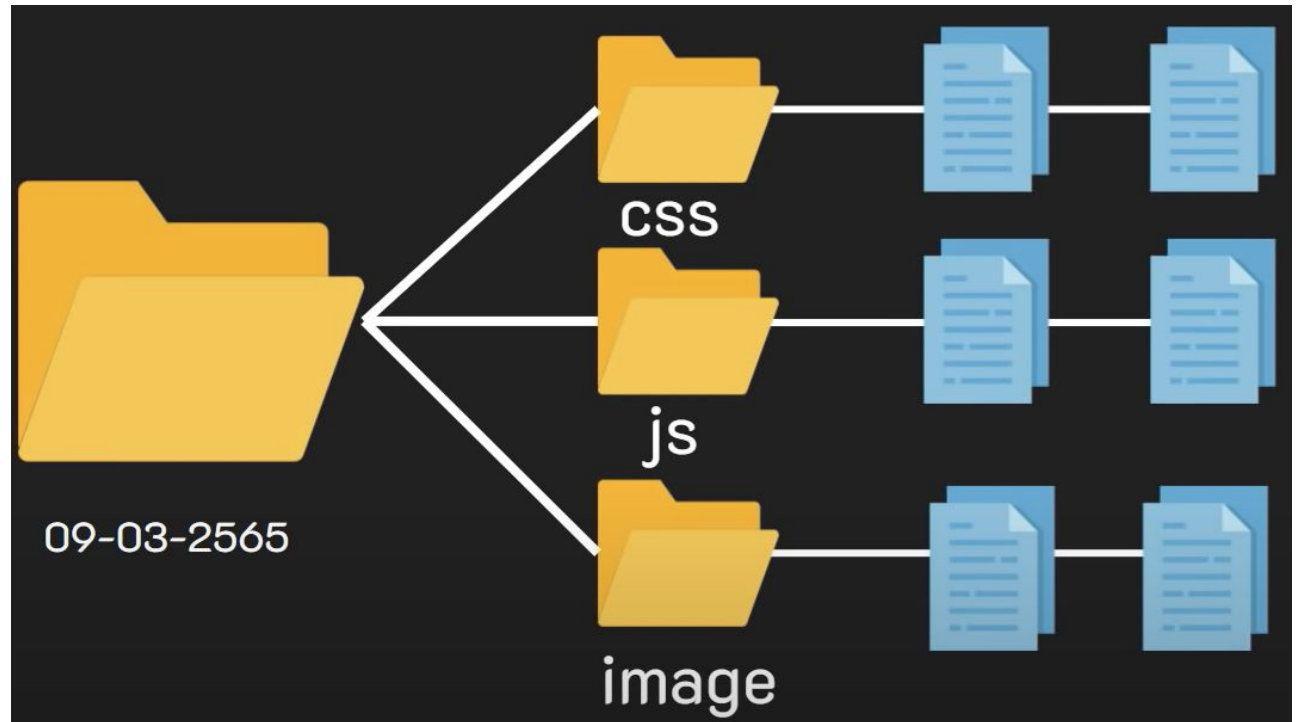
- ต้องส่งงานที่ทำล่าสุดให้กันอย่างไร ?
- คนในทีมทำการปรับปรุงแก้ไขโค้ดอะไรไปบ้าง ?
- ถ้าแก้ไขโค้ดจุดเดียวกันจะเกิดปัญหาหรือไม่ ?
- อยากพัฒนาระบบที่มีอยู่ไม่ให้กระทบกับ Production จะทำอย่างไร ?

วิวัฒนาการของ Version Control

- Copy File & Folder**
- Patch**
- Local Version Control System**
- Centralized Version Control System**
- Distributed Version Control System**

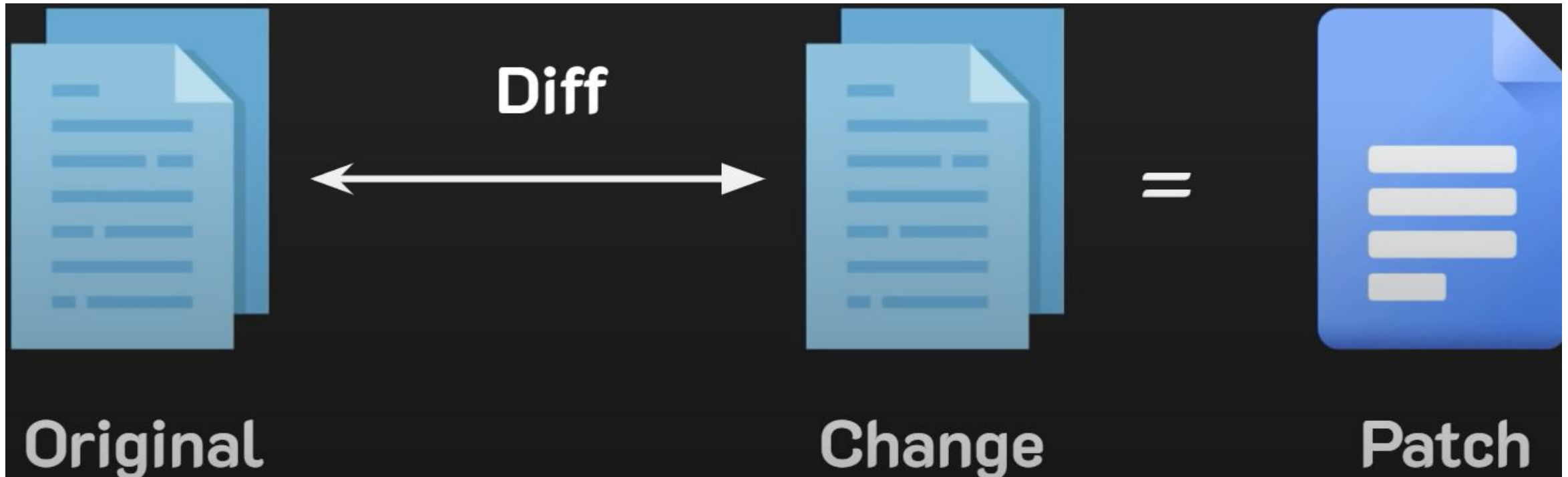
Copy File & Folder

	project.py	07-03-2565
	projectv1.py	10-03-2565
	projectv1.2.py	15-03-2565
	projectfinal.py	30-03-2565



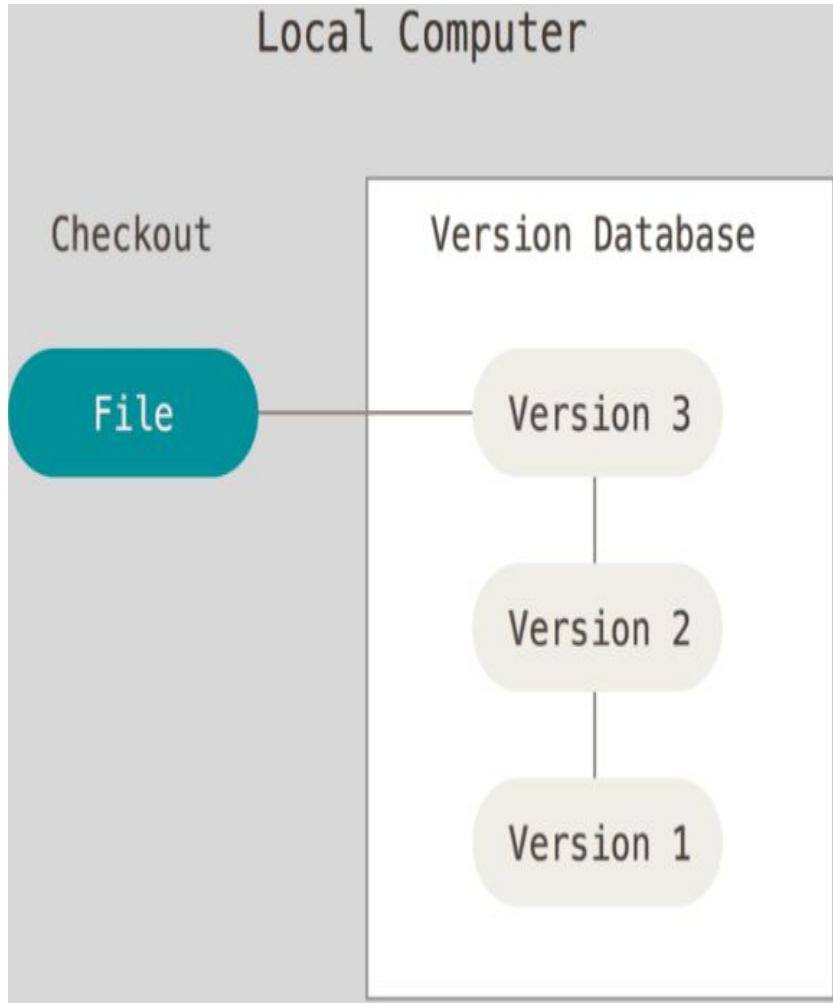
ข้อเสีย: การจัดเก็บไฟล์หรือโฟลเดอร์จำนวนมาก

Patch - **ลดเนื้อที่**ในการจัดเก็บไฟล์หรือไฟล์เดอ อร์จำนวนมาก



Local Version Control System

Where?



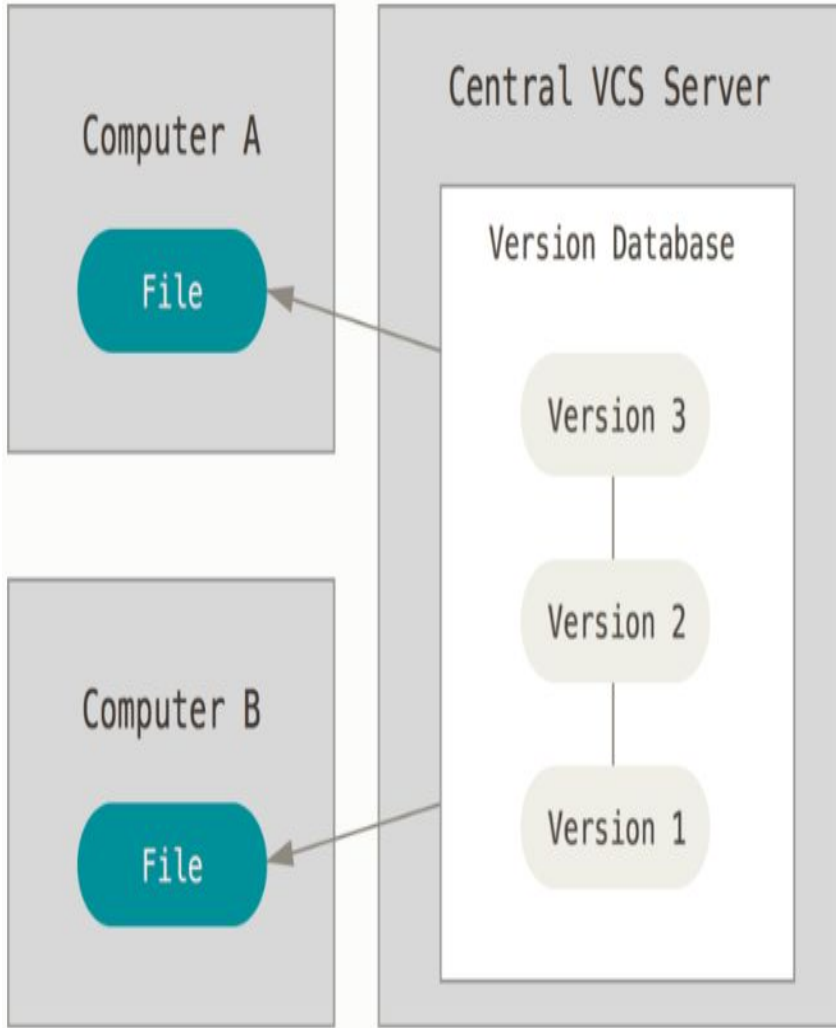
Check in: จัดเก็บเวอร์ชัน

Check out: เรียกคืนเวอร์ชันจาก

ข้อเสีย: check out จากหลาย

คน

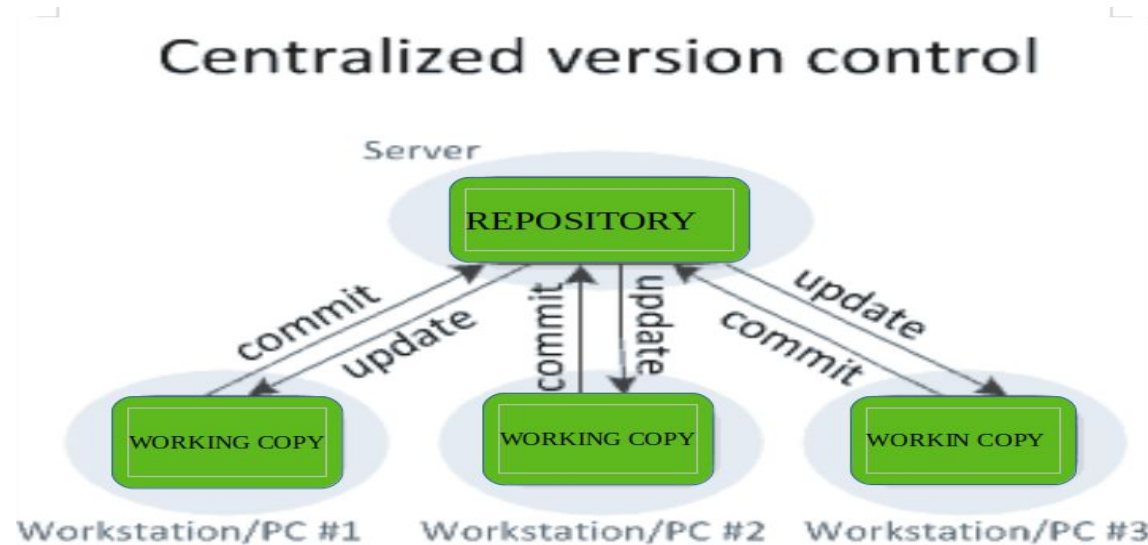
Centralized Version Control System



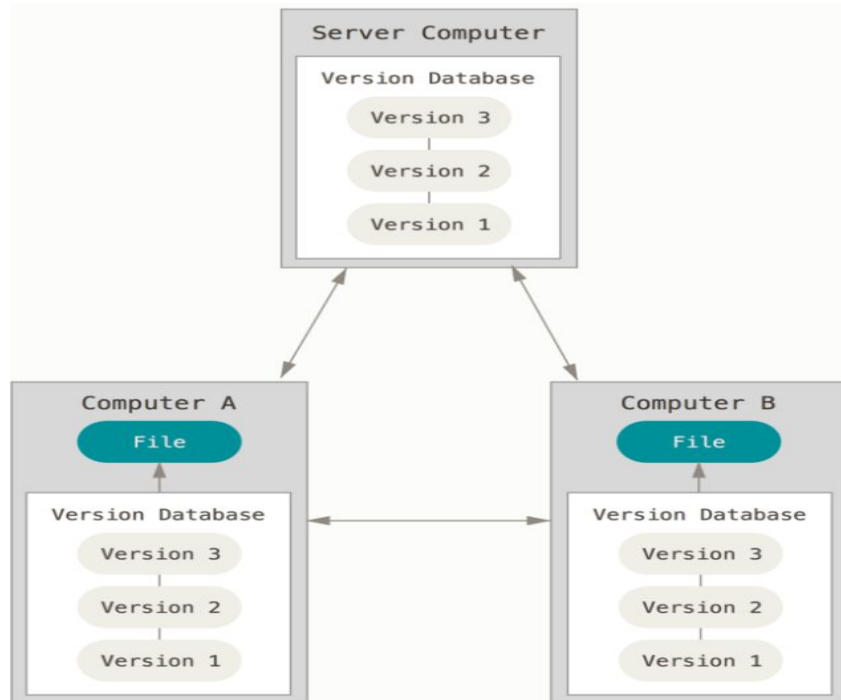
Check in: จัดเก็บเวอร์ชัน **Where?**

Check out: เรียกคืนเวอร์ชันจาก

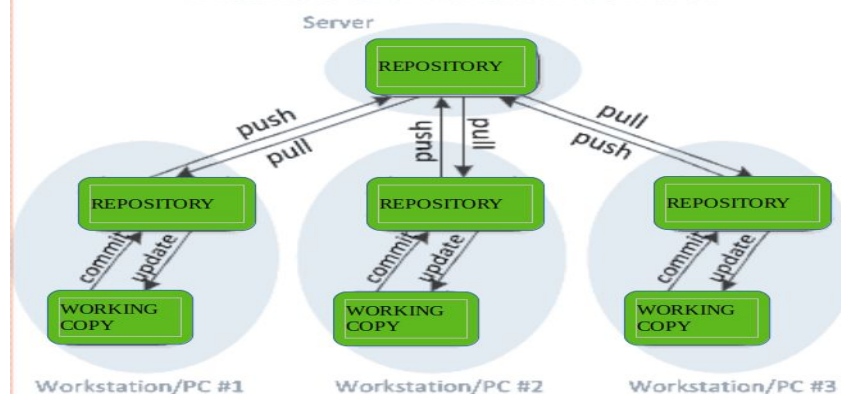
ข้อเสีย: เครื่อง Server เสีย
/Net



Distributed Version Control System



Distributed version control



Check in: จัดเก็บเวอร์ชัน
Check out: เรียกคืน
เวอร์ชันจากฐานข้อมูล

-- Local/Server --

Git เป็นแบบไหน?

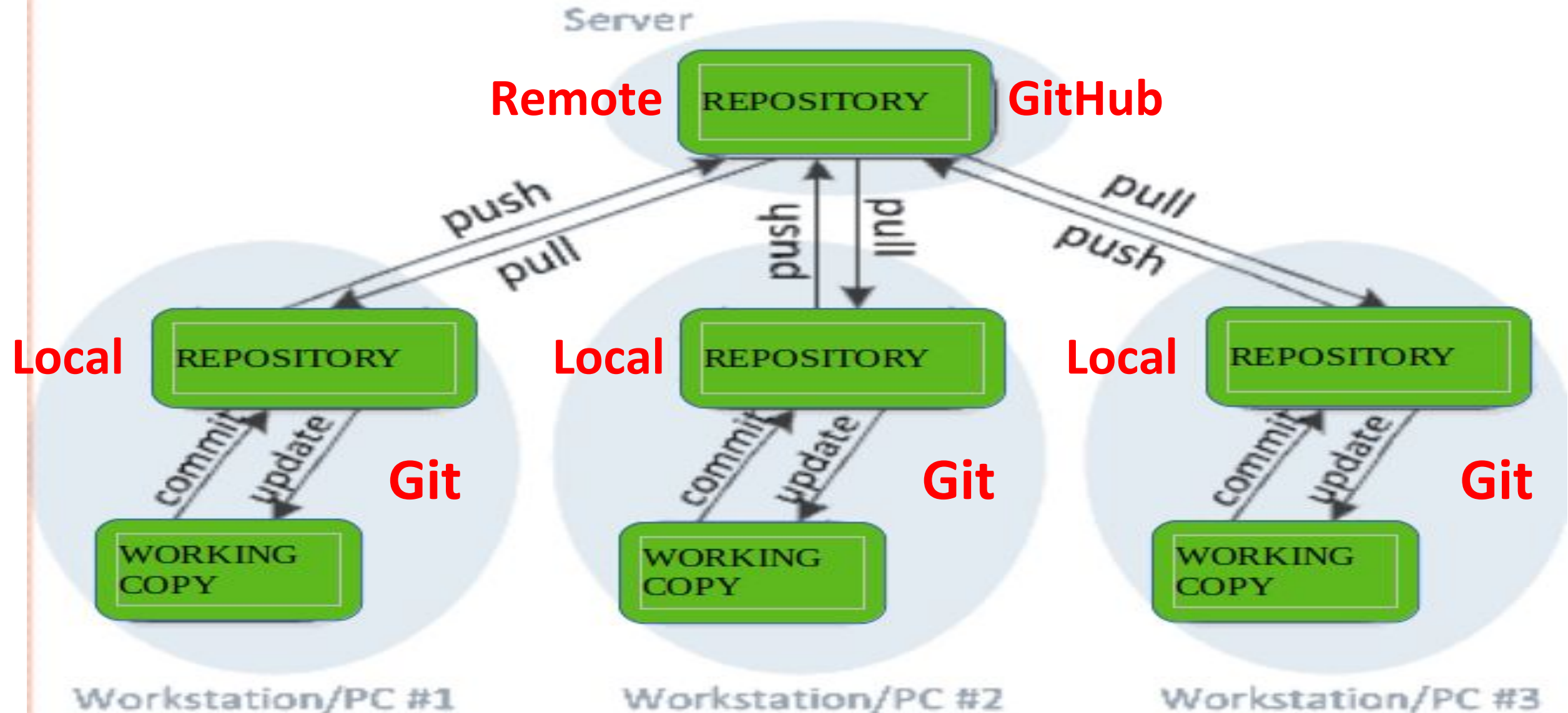
GitHub เป็นแบบไหน?

Git+GitHub เป็นแบบ

ไหน?

<https://www.geeksforgeeks.org/version-control-systems/>

Distributed version control



ทำไมต้องใช้ Git & GitHub

- ติดตามเวอร์ชันของโค้ดได้ (Track version)
- ทำงานเป็นทีม (Teamwork)
- Git for window - <https://gitforwindows.org/> (ติดตั้งด้วย
ครบ...ขอร้อง..ครับ)
 - `git -version` #ตรวจสอบว่าติดตั้งได้สำเร็จ
- GitHub - <https://github.com/> (สมัครสมาชิกครบ...ขอร้อง..
ครับ)
- NPM - <https://nodejs.org/en/download> (แถม)

Quasar = Java script + Vue + Quasar framework



```
$ npm i -g @quasar/cli  
$ npm init quasar
```

Quasar CLI is the pride of Quasar Framework. You can seamlessly build:

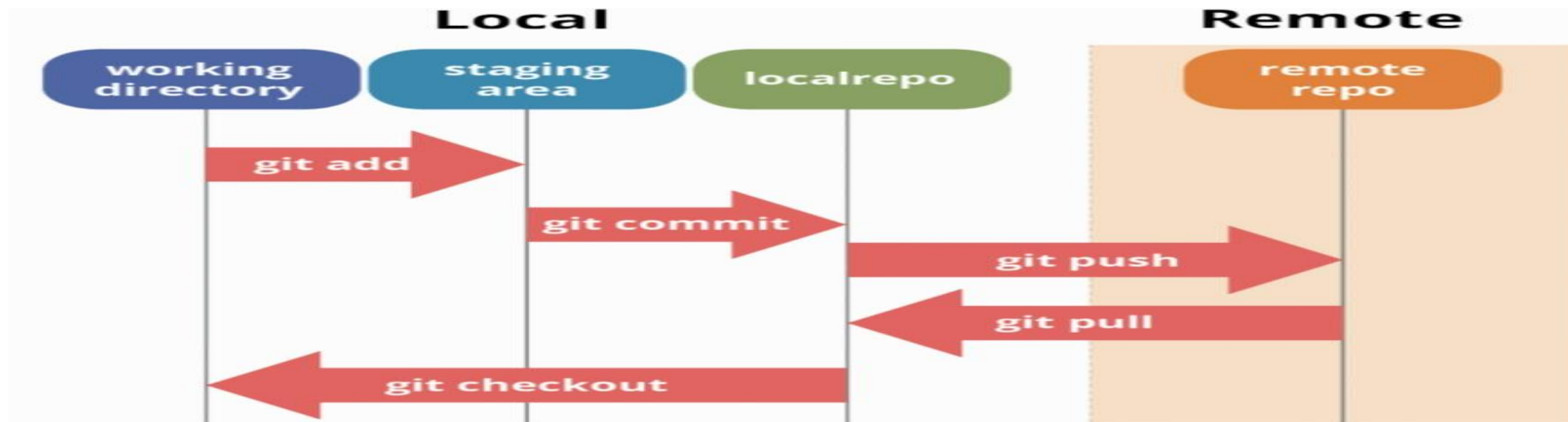
- a SPA (Single Page Application/Website),
- a SSR (Server-side Rendered App/Website),
- a PWA (Progressive Web App),
- a BEX (Browser Extensions),
- a Mobile App (through Cordova or Capacitor),
- an Electron App

Local Repository (Git bash)

- `git --version`
- `git config --list` #ใช้สำหรับตรวจสอบ config
- `git config -- global user.name = "name สมัครงาน GitHub"`
- `git config -- global user.email= "email สมัครงาน GitHub"`

```
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=somnuk@mju.ac.th
user.name=somnuk2
gui.recentrepo=D:/quasar-project
```

Git workflow



คำสั่งส่วน (local repository)

- git init #สร้าง local
- git add . #เพิ่มไฟล์เข้าไป local
- git status #ตรวจสอบสถานะของไฟล์
- git rm -r cached . #ลบไฟล์จาก local

git commit -m "ข้อความ" # ยืนยัน VER

git log #แสดงไฟล์ที่ได้ทำการยืนยัน

git log --oneline #แสดงไฟล์ที่ได้ทำการยืนยันแบบสั้น

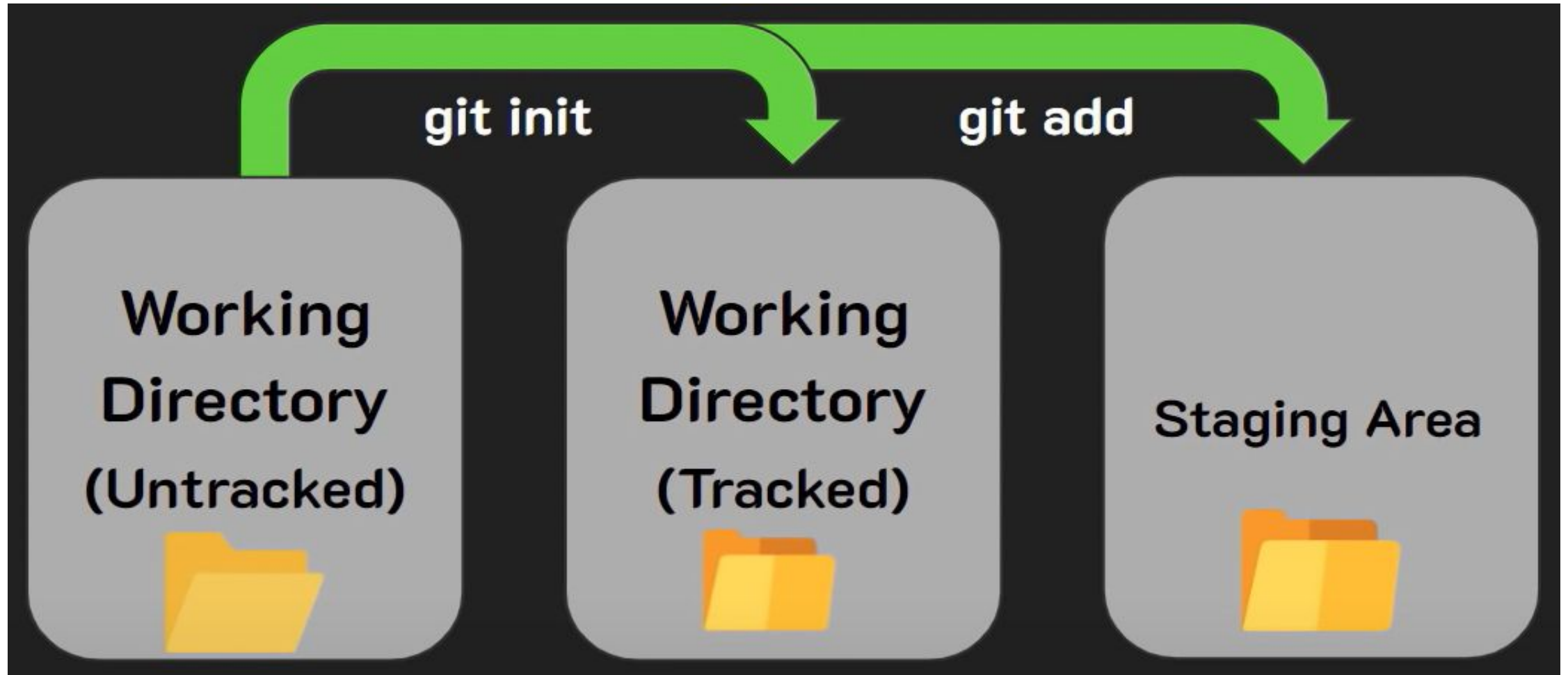
git log --graph #แสดงไฟล์ที่ได้ทำการยืนยันแบบกราฟ

git checkout <file> ยกเลิกแก้ไขไฟล์

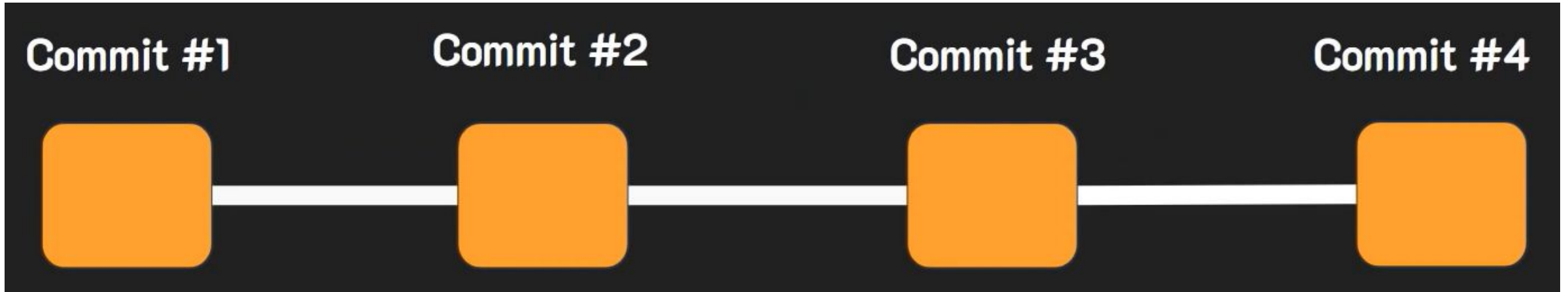
git diff <commit id> 1 <commit id> 2



`git reset <file>` – ย้อน file จาก add /staging area



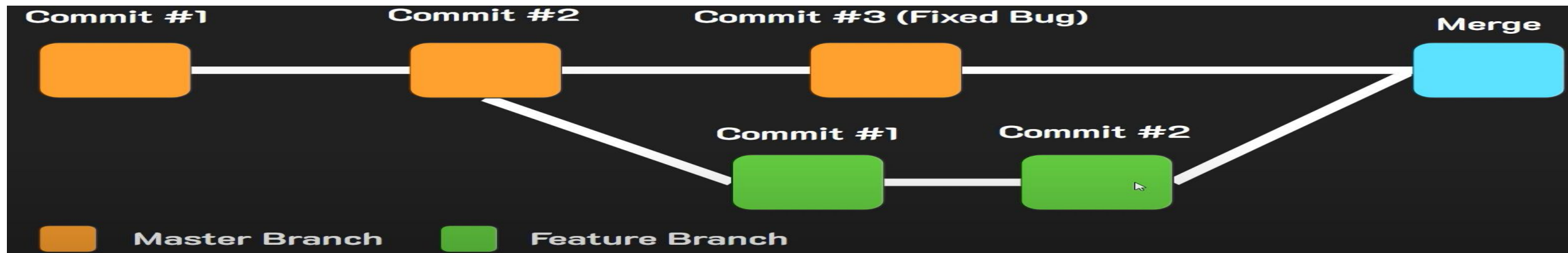
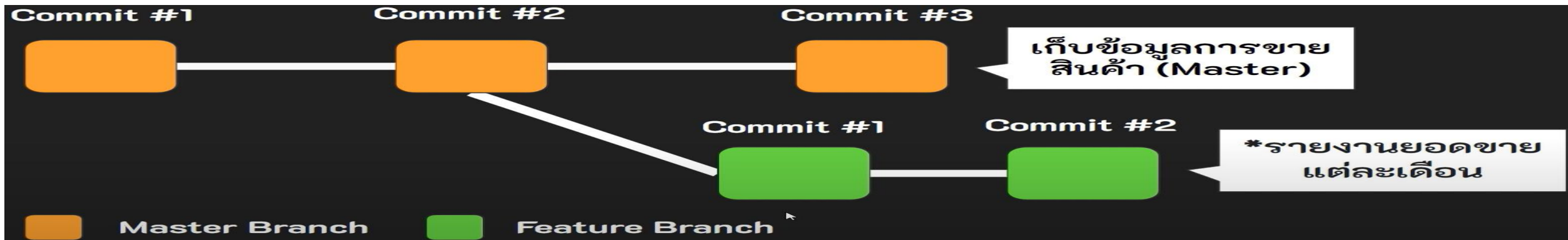
git reset <commit id/VER> – ย้อน VER



git reset -- option <commit_id>

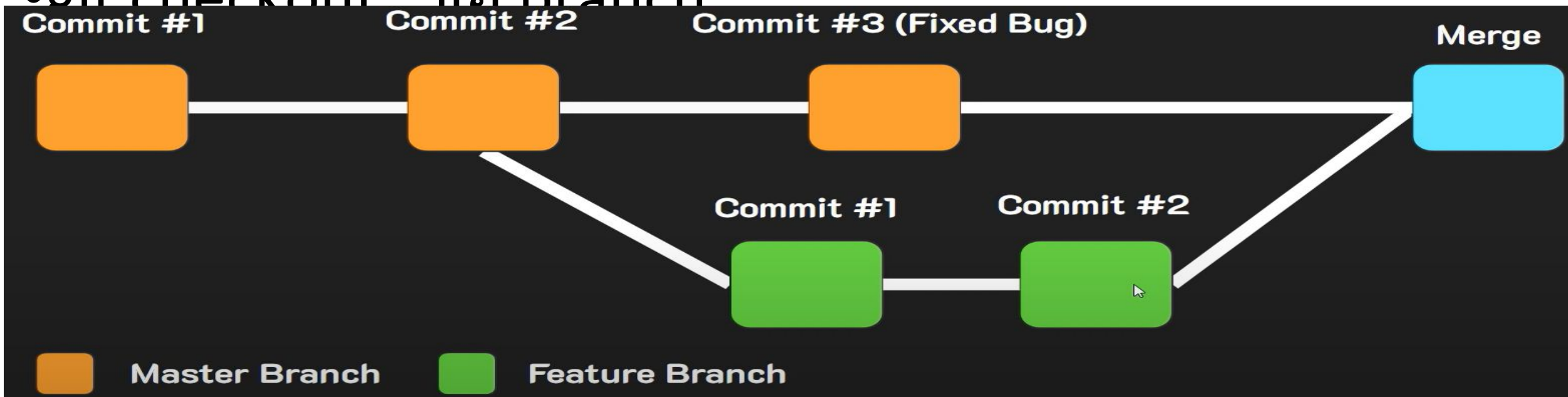
- **soft** ใช้เพื่อลบ Commit ทั้งหมดที่อยู่หลัง Commit ID แล้วนำไฟล์ที่เคยอยู่ใน Commit นั้นกลับมายัง Staging Area
- **mixed** ใช้เพื่อลบ Commit ทั้งหมดที่อยู่หลัง Commit ID แล้วนำไฟล์ที่เคยอยู่ใน Commit นั้นกลับมายัง Working Directory
- **hard** ใช้เพื่อลบ Commit ทั้งหมดที่อยู่หลัง Commit ID และจะทำลายไฟล์ที่เคยอยู่ใน Commit เหล่านั้น

Git branch

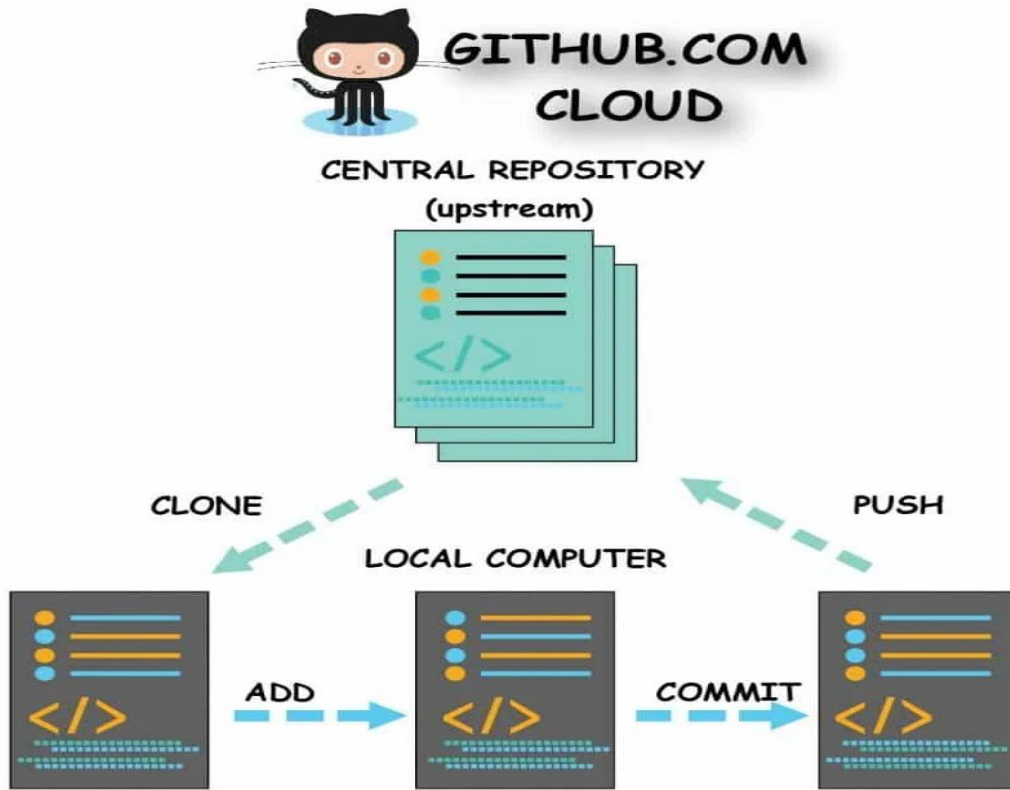


Git branch

- git branch
- git branch - b "ชื่อ branch"
- git checkout "ชื่อ branch"
- git branch - d "ชื่อ branch"
- git merge "ชื่อ branch"



Git push



<https://www.toolsqa.com/git/git-push/>