



เรียนรู้การใช้งาน

Git & GitHub

สำหรับผู้เริ่มต้น 2022



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



รู้จักกับการจัดเก็บเวอร์ชัน (Version Control)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Version Control คืออะไร

Version Control หรือ Source Control หมายถึง เครื่องมือช่วยติดตามการเปลี่ยนแปลงของ Source Code โดยการเก็บประวัติการเปลี่ยนแปลงลงฐานข้อมูลชนิดพิเศษ ซึ่งจุดประสงค์ของการเก็บบันทึกทุกการเปลี่ยนแปลงก็คือ หากมีความผิดพลาดเกิดขึ้น ไม่ว่าจะผิดพลาดเล็กน้อย ไปจนถึงขั้นร้ายแรงที่จะส่งผลให้ซอฟต์แวร์ที่พัฒนาขึ้นมาขึ้นพังทั้งหมด





Version Control คืออะไร

การบันทึกประวัติเก็บไว้อย่างต่อเนื่อง จะช่วยทำให้นักพัฒนาซอฟต์แวร์สามารถย้อนกลับไปในช่วงเวลาต่างๆ โดยเปรียบเทียบกับโค้ดเวอร์ชันก่อนหน้าได้ และตรวจสอบความผิดพลาดเพื่อแก้ไขให้มีผลกระทบต่อทีมพัฒนาน้อยที่สุด คล้ายๆกับการ Undo / Redo ในโปรแกรม (แต่ถ้าหากปิดโปรแกรมไปก็จะไม่สามารถ Undo / Redo อีกรอบได้)



BREAK!



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



สาเหตุที่ต้องใช้ Version Control

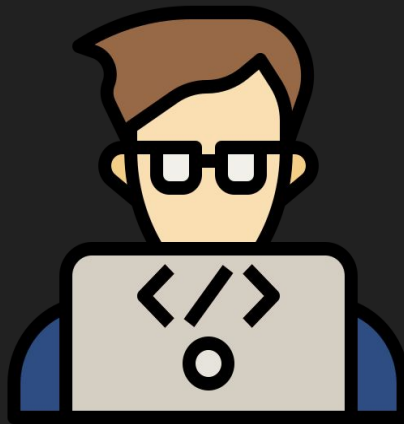


<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

สาเหตุที่ต้องใช้ Version Control



เด็กชายก้อง

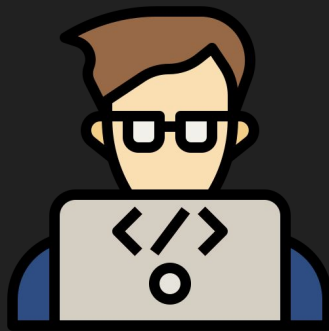


<https://www.youtube.com/c/kongruksiamofficial>



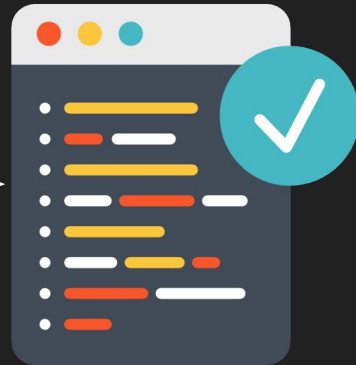
<https://www.facebook.com/KongRuksiamTutorial/>

สาเหตุที่ต้องใช้ Version Control

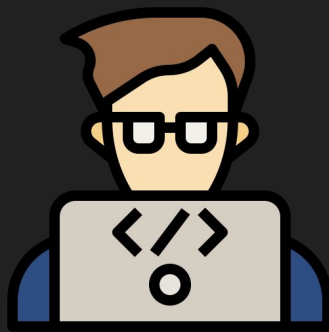


เด็กชายก้อง

เขียนโค้ด

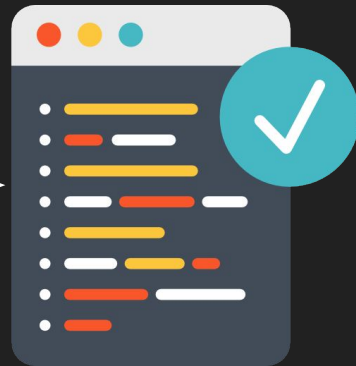


สาเหตุที่ต้องใช้ Version Control



เด็กชายก้อง

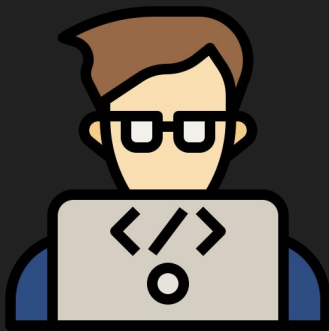
เขียนโค้ด



Source Code

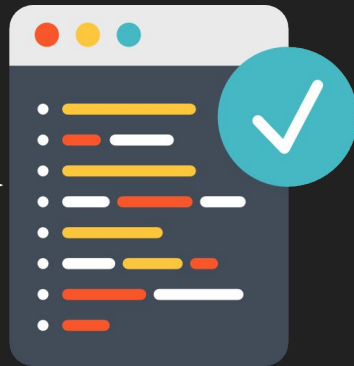


สาเหตุที่ต้องใช้ Version Control



เด็กชายก้อง

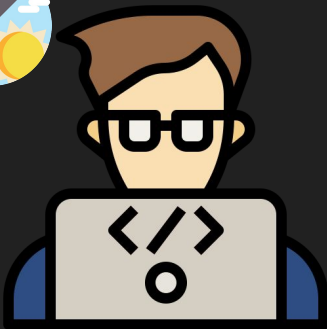
เขียนโค้ด



project

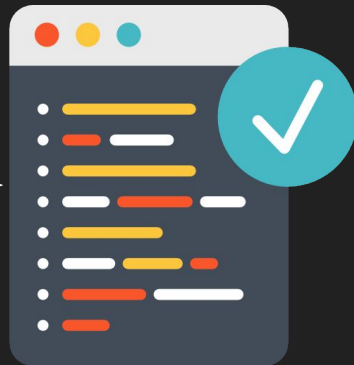


สาเหตุที่ต้องใช้ Version Control



เด็กชายก้อง

เขียนโค้ด



project

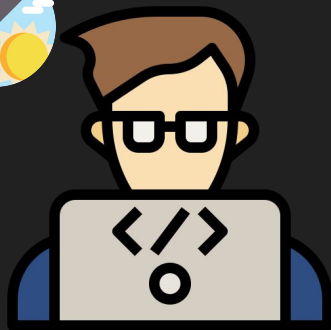


<https://www.youtube.com/c/kongruksiamofficial>

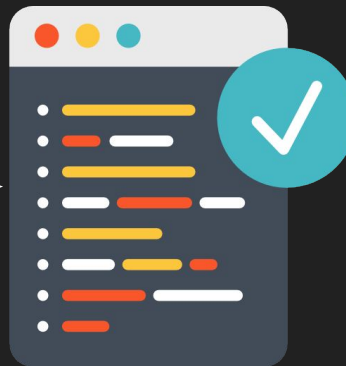


<https://www.facebook.com/KongRuksiamTutorial/>

สาเหตุที่ต้องใช้ Version Control



เด็กชายก้อง



project

- เก็บข้อมูลการขายสินค้า

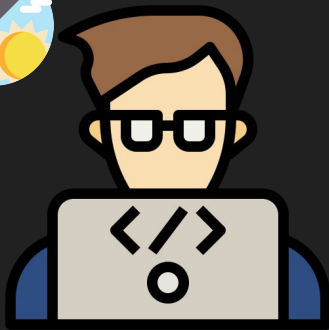


<https://www.youtube.com/c/kongruksiamofficial>

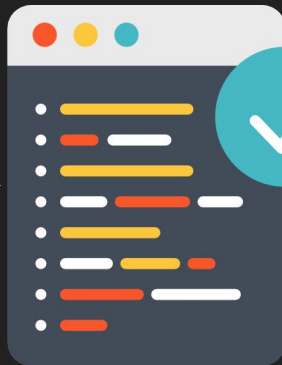


<https://www.facebook.com/KongRuksiamTutorial/>

สาเหตุที่ต้องใช้ Version Control

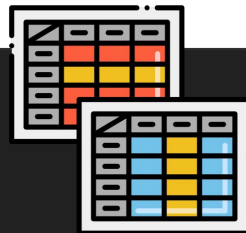


เด็กชายก้อง



project

- เก็บข้อมูลการขายสินค้า

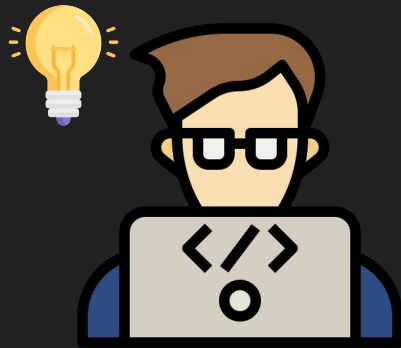


<https://www.youtube.com/c/kongruksiamofficial>

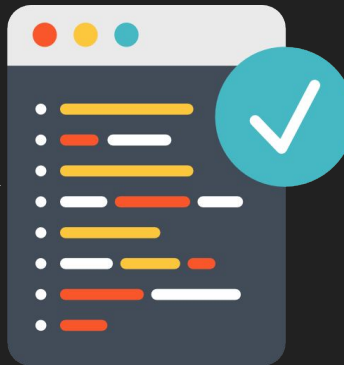


<https://www.facebook.com/KongRuksiamTutorial/>

สาเหตุที่ต้องใช้ Version Control

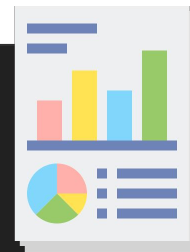


เด็กชายก้อง



project

- เก็บข้อมูลการขายสินค้า
- ****รายงานยอดขายแต่ละเดือน**

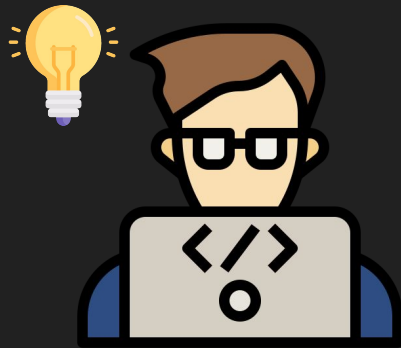


<https://www.youtube.com/c/kongruksiamofficial>

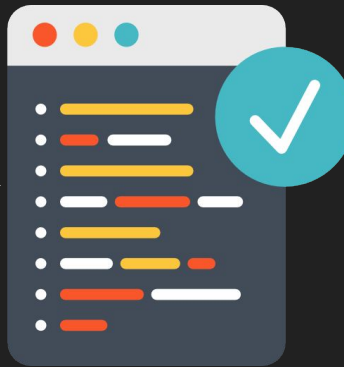


<https://www.facebook.com/KongRuksiamTutorial/>

สาเหตุที่ต้องใช้ Version Control



เด็กชายก้อง

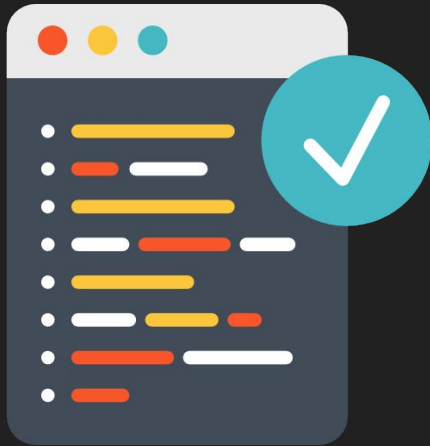


project

ทำยังไงจึงจะสามารถพัฒนา
โปรเจกต์ใหม่ / เพิ่มฟีเจอร์เข้าไป
ในโปรเจกต์เก่าและใช้งานระบบใน
เวอร์ชันเก่าได้ด้วย ?

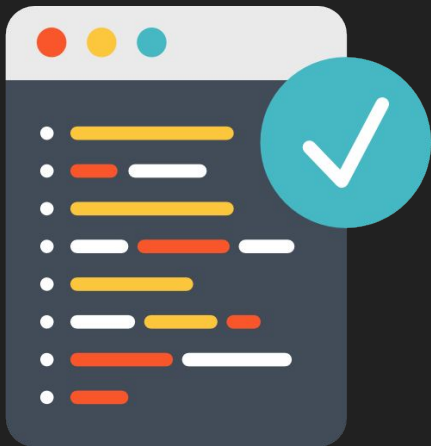


สาเหตุที่ต้องใช้ Version Control



1. Backup (สำรอง)โค้ดที่เขียนเพื่อให้สามารถถกกลับมาแก้ไขได้ในภายหลังได้

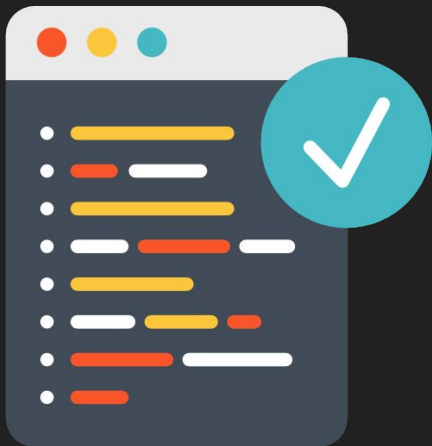
สาเหตุที่ต้องใช้ Version Control



2. คัดลอกไฟล์โค้ดเก่าและสร้าง
ไฟล์ใหม่ทุกครั้งที่มีการแก้ไขโค้ด



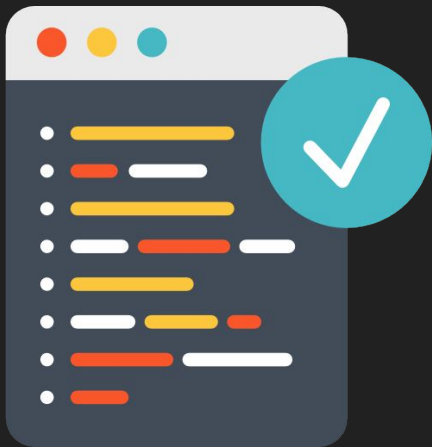
สาเหตุที่ต้องใช้ Version Control



- project



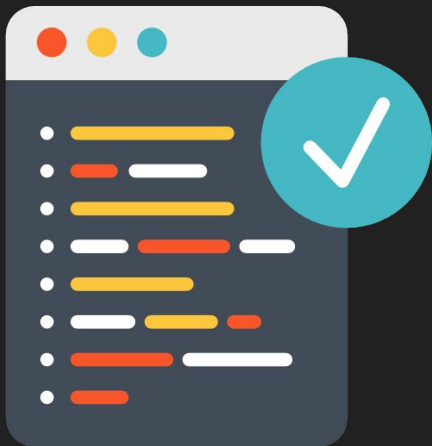
สาเหตุที่ต้องใช้ Version Control



- project
- projectv2



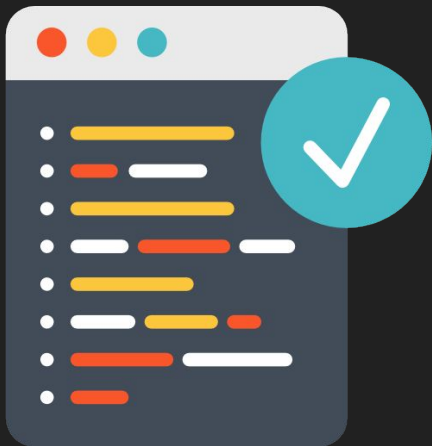
สาเหตุที่ต้องใช้ Version Control



- project
- projectv2
- projectv3 fixbug



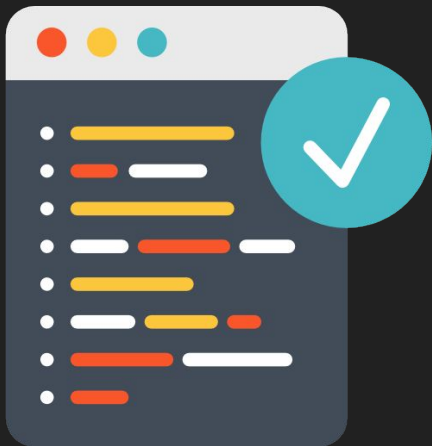
สาเหตุที่ต้องใช้ Version Control



****** เก็บโค้ดเวอร์ชันเก่าไว้เพื่อเวอร์ชันใหม่
ที่กำลังพัฒนานั้นเกิดพังขึ้นมา จะได้กลับ
ไปใช้ โค้ดเวอร์ชันเก่าได้นั่นเอง



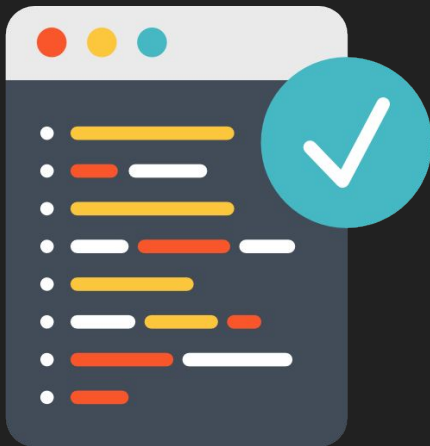
สาเหตุที่ต้องใช้ Version Control



- project
- projectv2
- projectv3 fixbug
- projectv4 add features

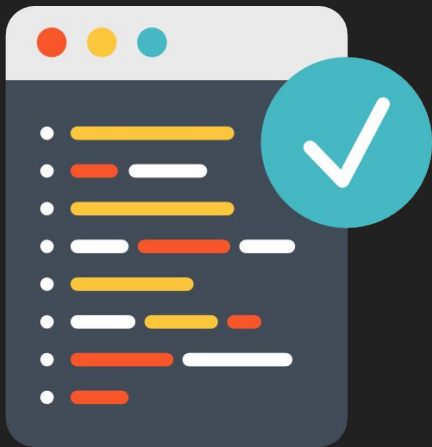


สาเหตุที่ต้องใช้ Version Control



- project
- projectv2
- projectv3 fixbug
- projectv4 add feature
- projectv5 final

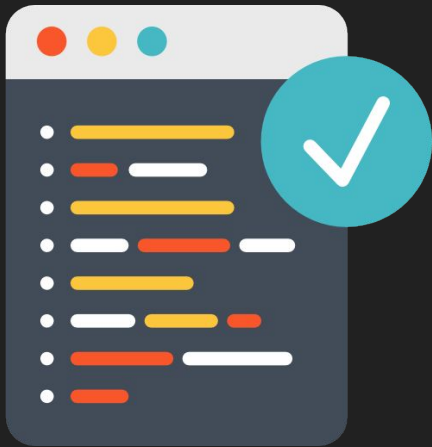
สาเหตุที่ต้องใช้ Version Control



- project
- projectv2
- projectv3 fixbug
- projectv4 add feature
- projectv5 final
- projectv5 final fixbug



สาเหตุที่ต้องใช้ Version Control



ถ้าเป็นโปรเจกต์ใหญ่
มีความซับซ้อน มีไฟล์เยอะๆล่ะ ?

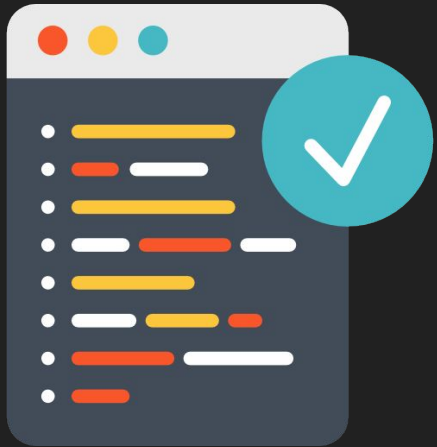


<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>

สาเหตุที่ต้องใช้ Version Control

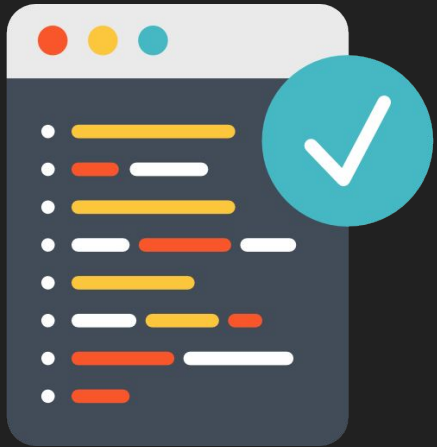


<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>

สาเหตุที่ต้องใช้ Version Control



07-03-2565

สาเหตุที่ต้องใช้ Version Control



07-03-2565



08-03-2565



09-03-2565



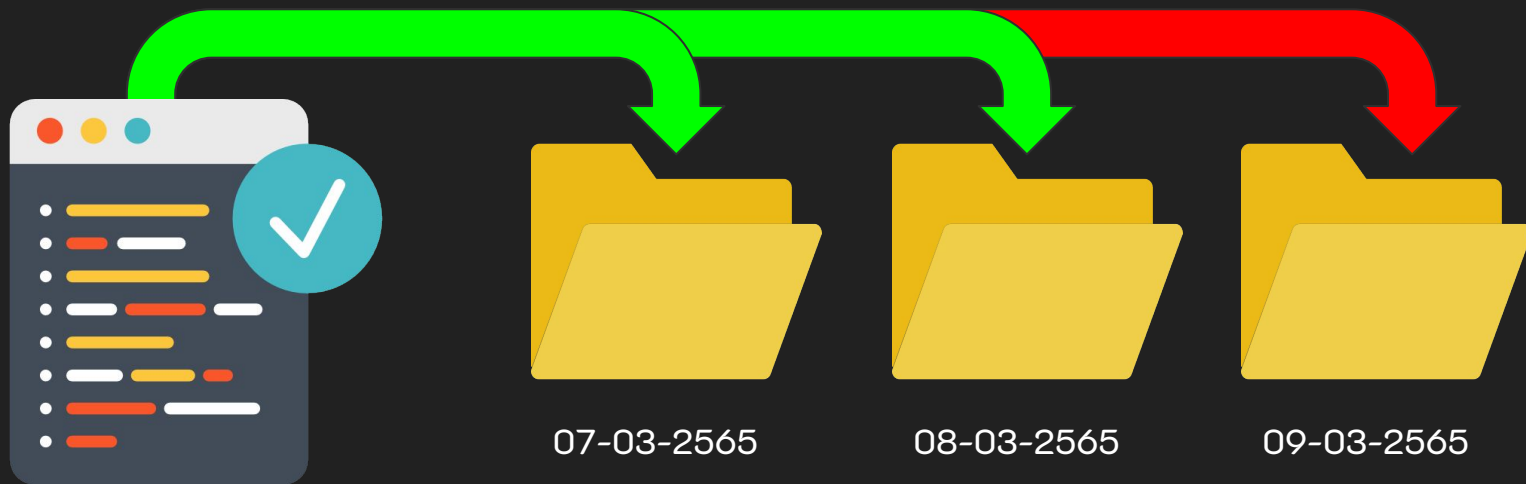
<https://www.youtube.com/c/kongruksiamofficial>



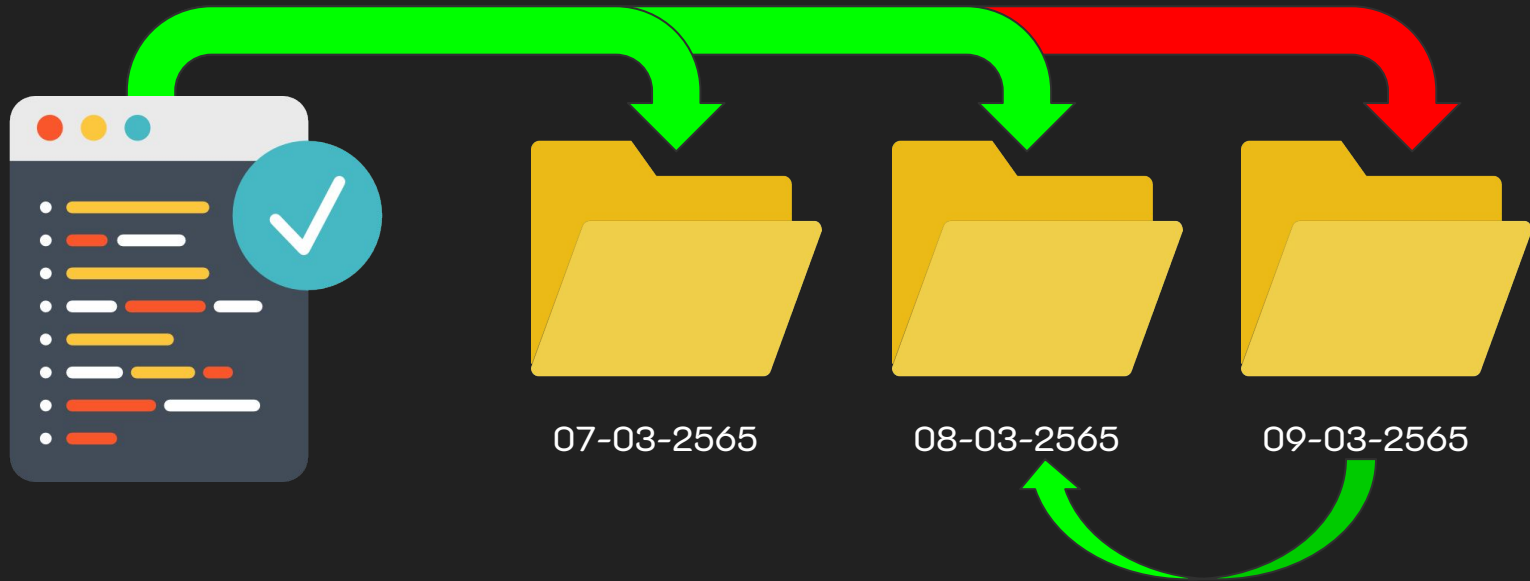
<https://www.facebook.com/KongRuksiamTutorial/>



สาเหตุที่ต้องใช้ Version Control



สาเหตุที่ต้องใช้ Version Control



สาเหตุที่ต้องใช้ Version Control

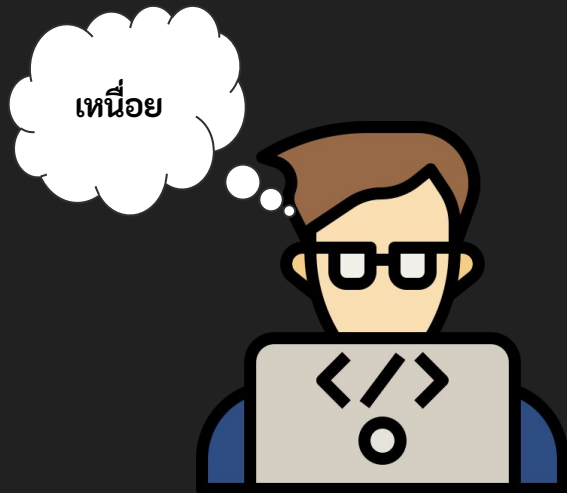


<https://www.youtube.com/c/kongruksiamofficial>



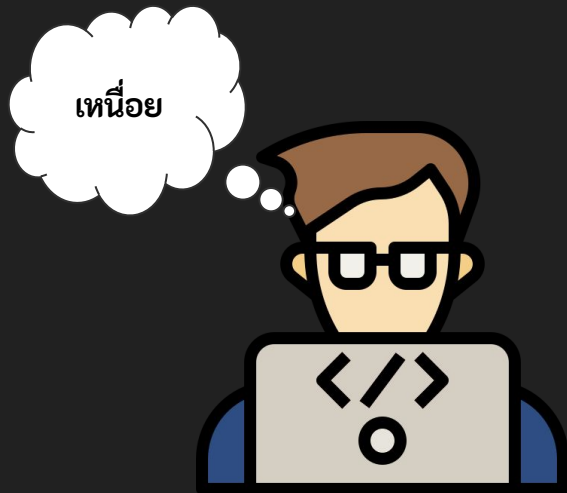
<https://www.facebook.com/KongRuksiamTutorial/>

สาเหตุที่ต้องใช้ Version Control



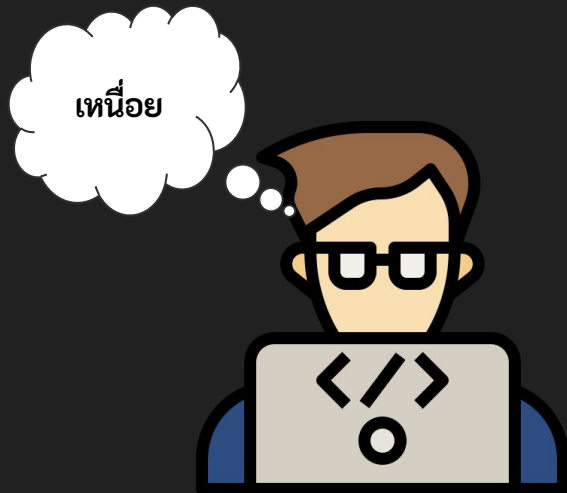
อยากกลับไปทำงานเก่าที่ทำไว้ จะต้อง
ไปหาไฟล์ Backup แล้วจำได้ไหมว่าไฟล์
Backup นั้น ถูกแก้ไขอะไรไปบ้าง
แก้ไขล่าสุดวันไหน แก้ไขเวลาใด ? / **ไม่รู้**

สาเหตุที่ต้องใช้ Version Control



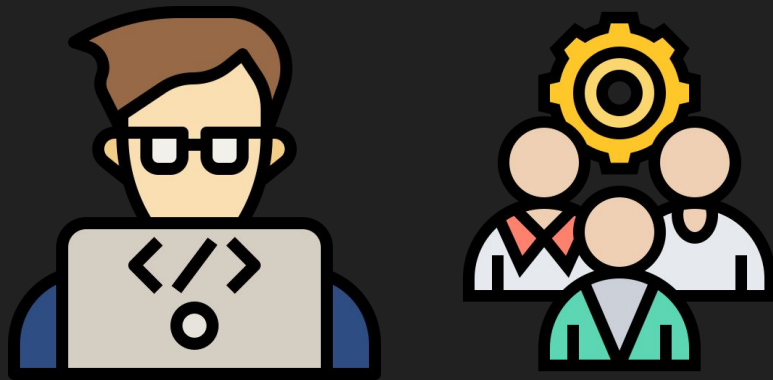
อยากทดลองเพิ่มฟีเจอร์ใหม่เข้าไป
ในโปรเจกต์เก่า จะทำอย่างไร
โดยที่ไม่ต้อง Copy โค้ดทั้งโปรเจ็ค
แล้วมาเปลี่ยนชื่อในภายหลัง / **ไม่รู้**

สาเหตุที่ต้องใช้ Version Control



หลังเขียนโค้ดเสร็จต้องทำการ Backup
โค้ดเก่าเก็บไว้ทุกวัน ถ้าวันหนึ่งพื้นที่เก็บ
ข้อมูลเต็มจะทำอย่างไร ? / **ไม่รู้**

สาเหตุที่ต้องใช้ Version Control

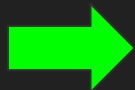
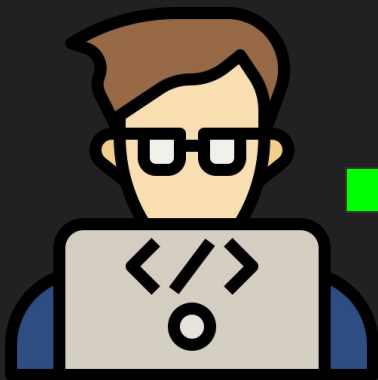


<https://www.youtube.com/c/kongruksiamofficial>

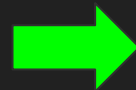


<https://www.facebook.com/KongRuksiamTutorial/>

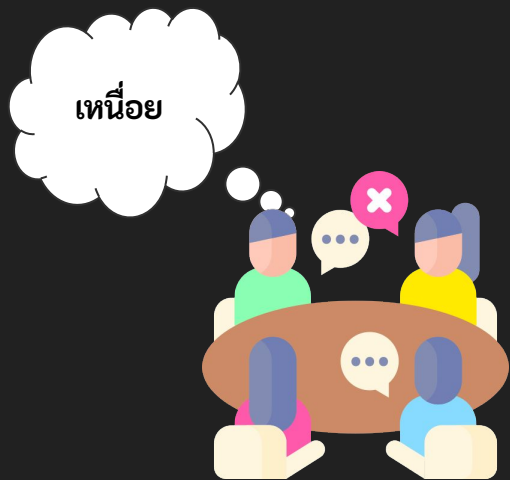
สาเหตุที่ต้องใช้ Version Control



- project
- projectv2
- projectv3 fixbug
- projectv4 add feature
- projectv5 final
- projectv5 final fixbug



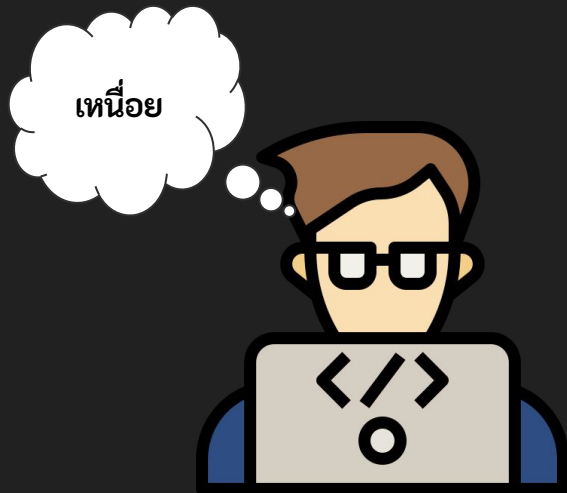
สาเหตุที่ต้องใช้ Version Control



- ต้องส่งงานที่ทำล่าสุดให้กันอย่างไร ?
- คนในทีมทำการปรับปรุงแก้ไขโค้ดอะไรไปบ้าง ?
- ถ้าแก้ไขโค้ดจุดเดียวกันจะเกิดปัญหาหรือไม่ ?
- อยากพัฒนาระบบที่มีอยู่ไม่ให้เกิดกระทบกับ Production จะทำอย่างไร ?



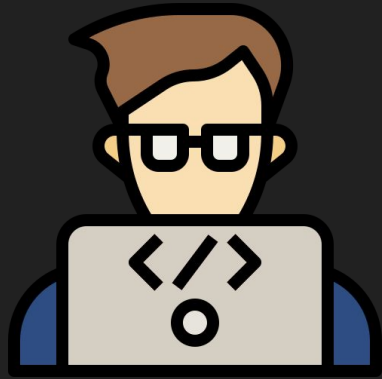
สาเหตุที่ต้องใช้ Version Control



- หาเครื่องมือมาช่วยในการเก็บการเปลี่ยนแปลงของไฟล์ได้ โดยการเก็บประวัติไฟล์ว่าถูกสร้าง/ลบ/แก้ไข โดยใคร เมื่อไหร่
- สามารถติดตามการเปลี่ยนแปลงของโค้ดในไฟล์ได้หรือย้อนเวลาโค้ดกลับไปก่อนตอนที่พังได้



สาเหตุที่ต้องใช้ Version Control



“ Version Control ”

BREAK!



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



วิวัฒนาการของ Version Control



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



วิวัฒนาการของ Version Control

1. Copy File & Folder
2. Patch
3. Local Version Control System
4. Centralized Version Control System (CVCS)
5. Distributed Version Control System (DVCS)





วิวัฒนาการของ Version Control

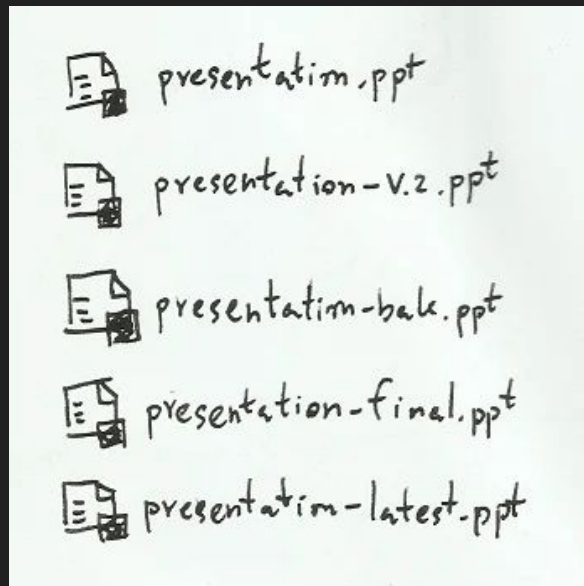
1. Copy File & Folder
2. Patch
3. Local Version Control System
4. Centralized Version Control System (CVCS)
5. Distributed Version Control System (DVCS)





Copy File & Folder

คือ ลักษณะการบันทึก
เอกสารและแยกออกเป็นหลาย
ๆไฟล์ แล้วตั้งชื่อไฟล์พร้อมระบุ
เวอร์ชันตามลำดับ



<https://www.blognone.com/sites/default/files/externals/06af9600e2d49c14619e38cec7ed4905.jpg>

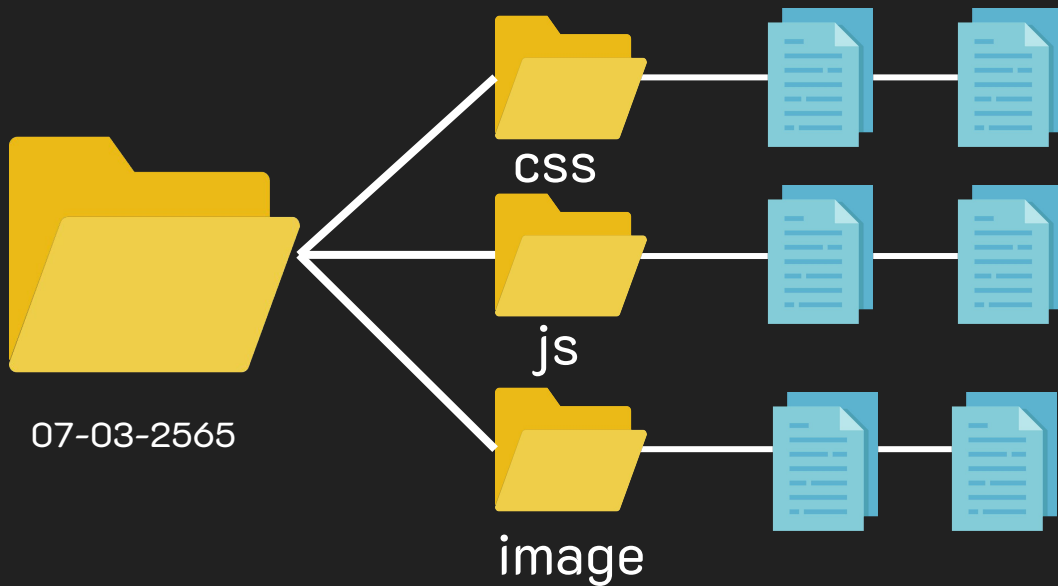


<https://www.youtube.com/c/kongruksiamofficial>

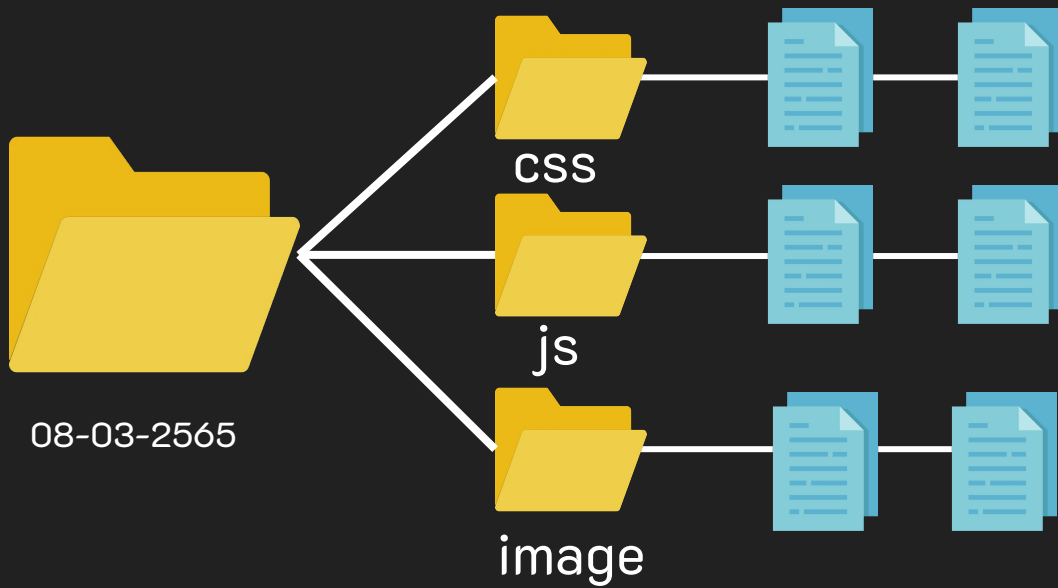


<https://www.facebook.com/KongRuksiamTutorial/>

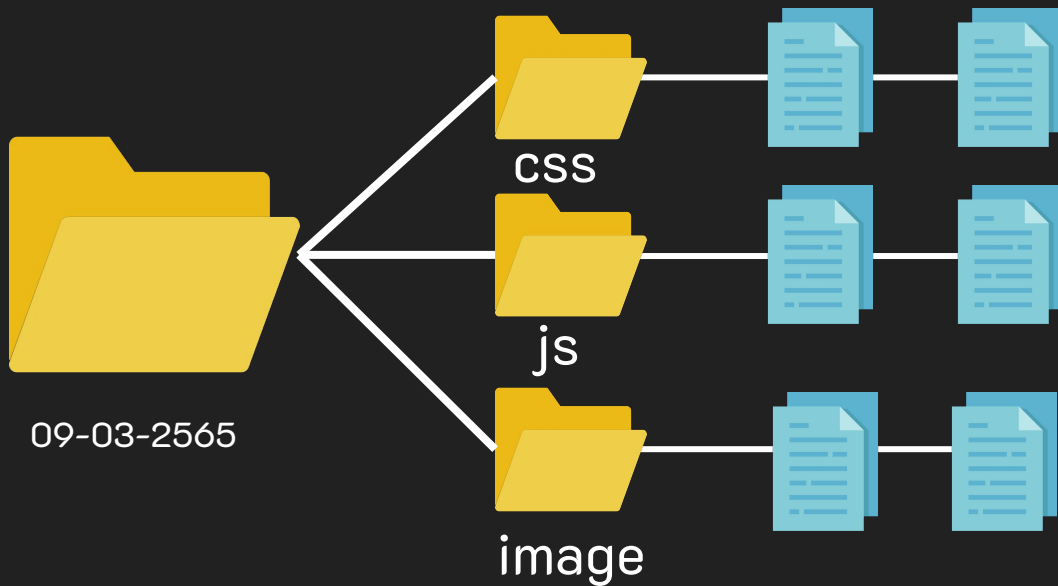
ตัวอย่าง



ตัวอย่าง



ตัวอย่าง



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>

ตัวอย่าง



07-03-2565



08-03-2565



09-03-2565



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>

ตัวอย่าง



project.py



07-03-2565



projectv1.py



10-03-2565



projectv1.2.py



15-03-2565



projectfinal.py



30-03-2565



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Copy File & Folder

**ข้อเสีย

เปลืองพื้นที่จัดเก็บข้อมูลตามจำนวนเวอร์ชันของเอกสาร เพราะต้องคัดลอกไฟล์ทั้งโฟลเดอร์เพื่อสร้างเอกสารเวอร์ชันใหม่ขึ้นมา



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



วิวัฒนาการของ Version Control

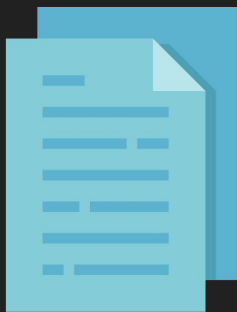
1. Copy File & Folder
2. Patch
3. Local Version Control System
4. Centralized Version Control System (CVCS)
5. Distributed Version Control System (DVCS)



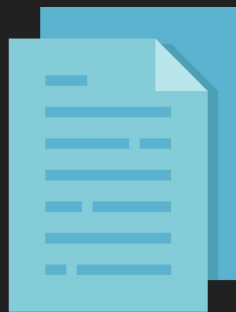


Patch (แพตช์)

Patch (แพตช์) เพื่อเล่ายความเปลี่ยนแปลงพื้นที่ในการจัดเก็บแบบวิธี Copy File & Folder เกิดจากการเปรียบเทียบไฟล์เก่า กับไฟล์ใหม่ที่มีการเปลี่ยนแปลง

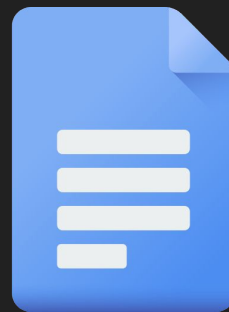


Original



Change

=



Patch

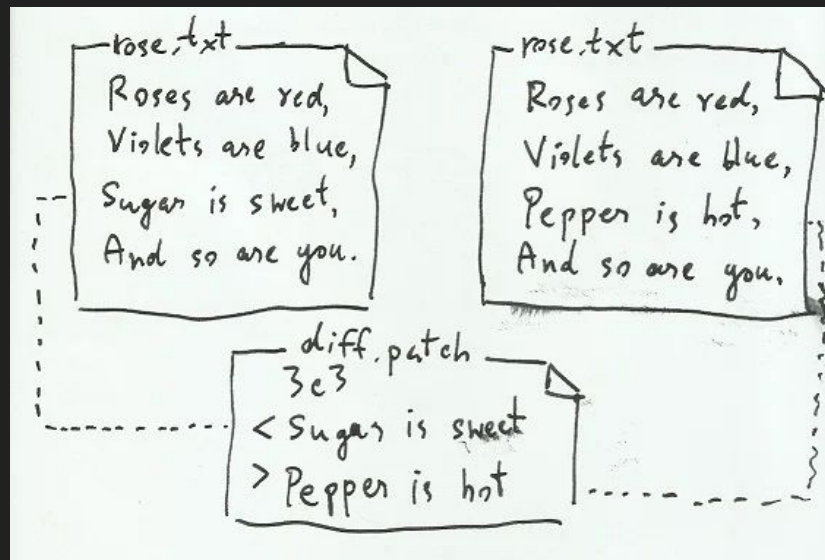


Patch (แพตช์)

**ข้อเสีย

อาจมีความเสี่ยงต่อการทำ

Patch บาง Patch หาย จนไม่สามารถประกอบร่าง Source Code
กลับไปยังเวอร์ชันต่างๆได้



<https://www.blognone.com/node/78730>



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



วิวัฒนาการของ Version Control

1. Copy File & Folder
2. Patch
3. Local Version Control System
4. Centralized Version Control System (CVCS)
5. Distributed Version Control System (DVCS)





Local Version Control System

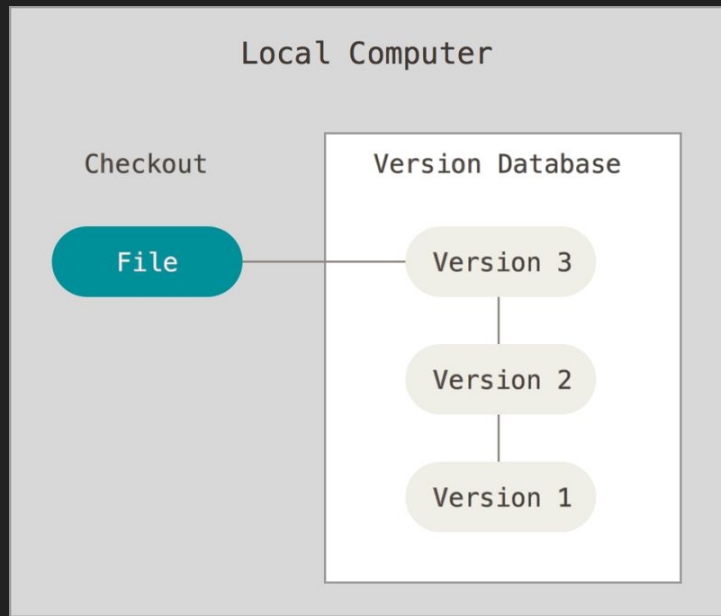


เพื่อแก้ปัญหา Patch หาย จนไม่สามารถประกอบร่าง Source Code กลับไปยังเวอร์ชันต่างๆได้ จึงได้มีการพัฒนา Version Control System (VCS) ที่มีฐานข้อมูลเฉพาะคอยจัดเก็บทุกการเปลี่ยนแปลงของ Source Code โดยจะเรียก**การจัดเก็บ**เวอร์ชันของ Source Code ลงฐานข้อมูลเรียกว่า “**Check-In**” และเรียกคืน Source Code จากฐานข้อมูลเพื่อทำงานต่อว่า “**Check-Out**”



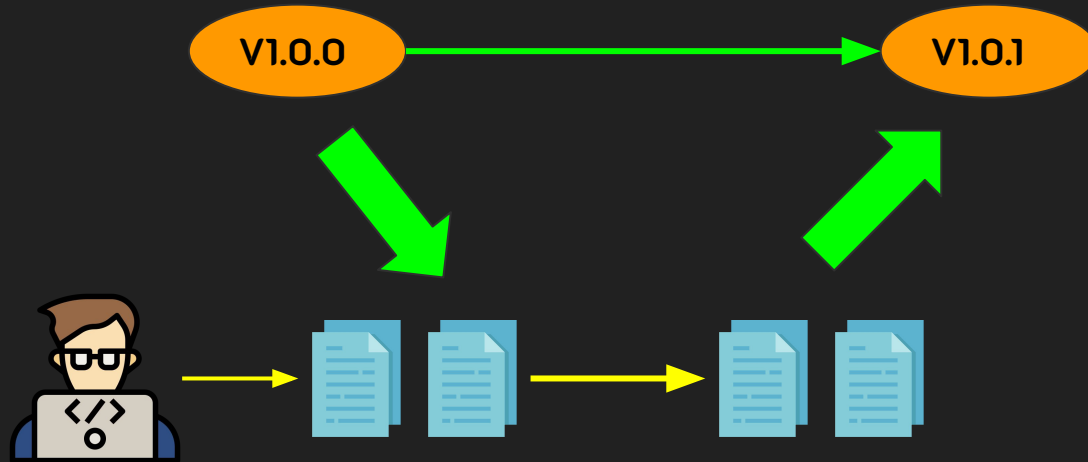
Local Version Control System

Local Version Control System มีความสามารถในการจัดเก็บเวอร์ชัน (Check-in) พร้อมทั้งข้อความช่วยจำ (Log Message) ลงในฐานข้อมูลและเรียกคืนเวอร์ชันจากฐานข้อมูล (Check-Out) กลับมายังพื้นที่ทำงาน เพื่อให้นักพัฒนาซอฟต์แวร์แก้ไขข้อผิดพลาดต่อไปได้



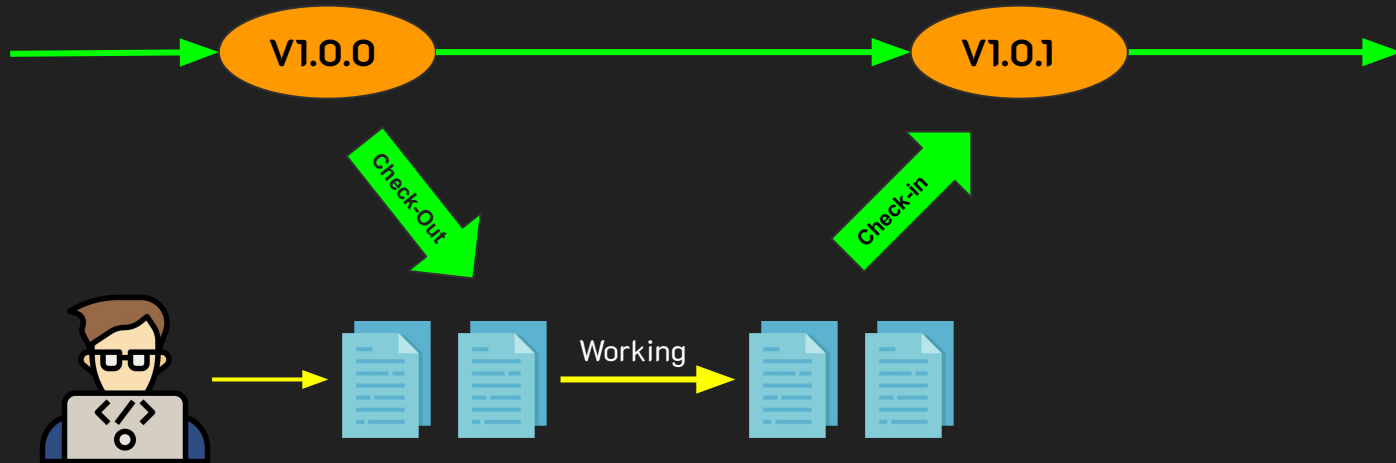


Local Version Control System



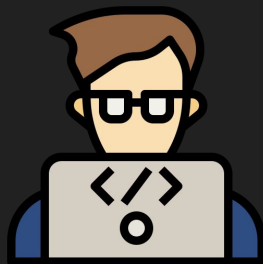


Local Version Control System





Local Version Control System



ข้อเสีย

การ Check-Out จากผู้ใช้หลายคน อาจจะทำให้เกิดปัญหาขึ้น



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



วิวัฒนาการของ Version Control

1. Copy File & Folder
2. Patch
3. Local Version Control System
4. Centralized Version Control System (CVCS)
5. Distributed Version Control System (DVCS)

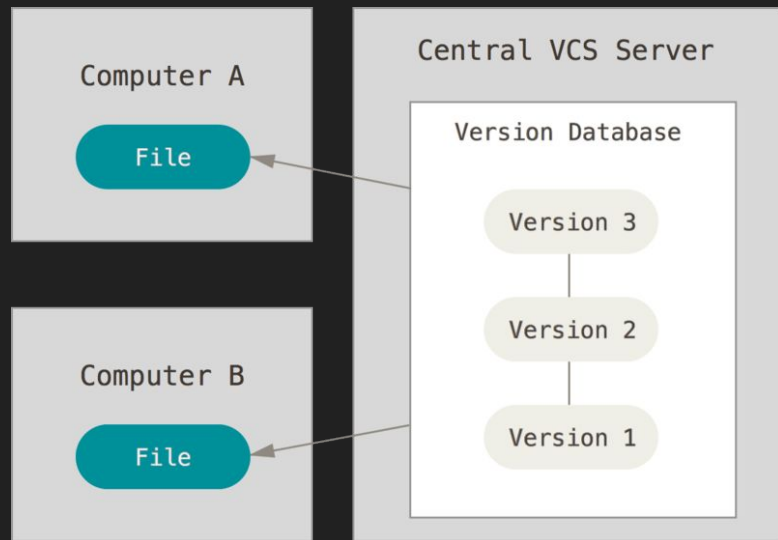




Centralized Version Control System

เป็น Version Control System แบบรวมศูนย์ เพื่อแก้ปัญหากรณีที่มีผู้ใช้งานหลายคนโดยการเก็บข้อมูลไว้บน Server ซึ่งเมื่อผู้ใช้ Check-Out งาน เวอร์ชันเดียวกันแล้วกลับมา Check-in ระบบจะพยายามรวบรวมเนื้อหาเข้าด้วยกัน (Merge)

- ถ้ารูปแบบการรวมเนื้อหามีความเรียบง่าย ระบบจะรวมเนื้อหาให้อัตโนมัติ
- ถ้าการรวมมีความซับซ้อน ระบบจะแจ้งให้ ผู้ใช้งานตัดสินใจแทน

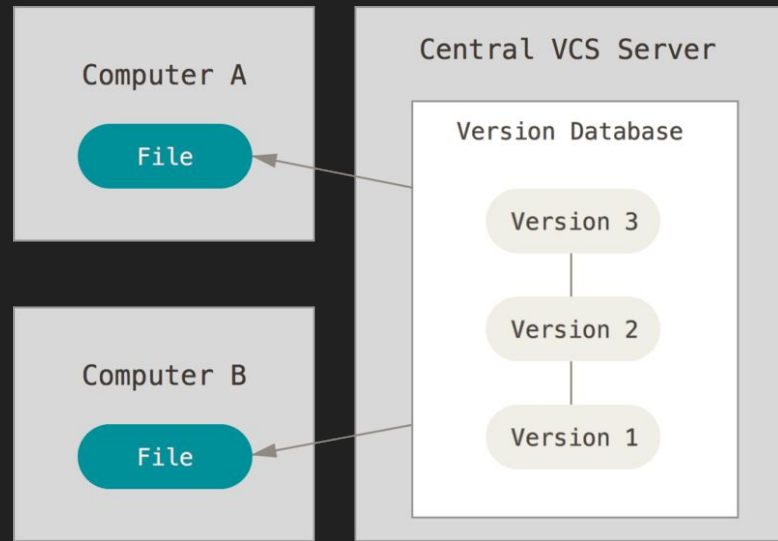




Centralized Version Control System

ข้อเสีย

- การทำงานแบบรวมศูนย์ หาก Server ล่มชั่วคราว ผู้ใช้จะไม่สามารถ Check-in หรือ Check-Out ได้
- ถ้า Server ล่มถาวร เวอร์ชันของ Source Code ทั้งหมดก็จะได้รับผลกระทบไปด้วย
- ทำงานแบบ Online เท่านั้น





วิวัฒนาการของ Version Control

1. Copy File & Folder
2. Patch
3. Local Version Control System
4. Centralized Version Control System (CVCS)
5. Distributed Version Control System (DVCS)





Distributed Version Control System

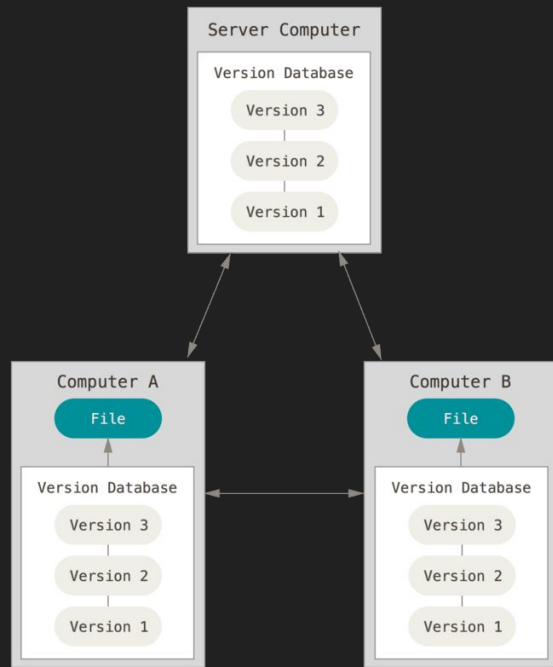
Distributed Version Control System (DVCS)

จะแก้ปัญหการทำงานแบบรวมศูนย์โดยการ
โคลนฐานข้อมูลมาทั้งหมด (เรียก Database
ที่เก็บเวอร์ชันของโค้ดว่า Repository) ซึ่งเมื่อ

Clone มาแล้ว สามารถ Check-In และ

Check-Out บน Local Host แบบ Offline

ก่อนจะ Push ขึ้น Server ในภายหลัง



สรุป



จุดประสงค์ที่สำคัญของการใช้งาน Version Control System คือ เพื่อให้สามารถย้อนกลับไปยังเวอร์ชันก่อนหน้าได้ เมื่อพบปัญหาระหว่างพัฒนาโปรแกรม การ Check-in เพื่อเปลี่ยนแปลง Source Code ไปยังระบบจัดเก็บเวอร์ชัน (Version Control System) จึงเป็นเรื่องที่ต้องทำเป็นประจำ

เพราะถ้าคนในทีมพัฒนาโปรแกรมไม่ทำการ Check-In เป็นประจำ แล้วพบปัญหาขึ้นกับซอร์ฟแวร์ที่อยู่ระหว่างพัฒนา ก็อาจก่อให้เกิดปัญหาใหญ่ตามมาคือ ไม่สามารถย้อนกลับไปยังเวอร์ชันต่างๆก่อนหน้าได้ หรืออาจจะต้องยุ่งยากหรือใช้เวลานานกว่าจะย้อนกลับไปได้

BREAK!



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



รู้จักกับ Git & GitHub



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ทำความรู้จักกับ Git



Git เป็น Version Control รูปแบบ
Distributed Version Control System
(DVCS) ใช้สำหรับติดตาม ตรวจสอบการ
แก้ไขไฟล์ หรือ Source Code ซึ่งสามารถ
ตรวจสอบได้ทุกตัวอักษร ทุกบรรทัด ทุกไฟล์
พร้อมทั้งระบุว่าไฟล์ดังกล่าวถูกปรับปรุง
แก้ไข ณ ช่วงเวลาใด และใครเป็นผู้แก้ไข

ทำความรู้จักกับ Git



ทำให้ผู้พัฒนาสามารถติดตาม
การเปลี่ยนแปลงของโค้ดได้ตลอด
หรือแม้กระทั่งย้อนเวลาโค้ดกลับ
ไปก่อนตอนที่จะพังก็ยังสามารถทำได้



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ทำความรู้จักกับ Git



ดังนั้น Git จึงเหมาะสำหรับนักพัฒนาไม่ว่าจะเป็นงานแบบเดี่ยวหรือแบบทีมเพราะสามารถเรียกดูได้ ว่าคนในทีมใครเป็นคนเขียนหรือใครเป็นคนแก้ไขโปรเจกต์ในแต่ละส่วน



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

หลักการพื้นฐานของ Git



สามารถคัดลอกเวอร์ชันของ Source Code มาเก็บไว้ในเครื่องก่อนได้ (Local Host) ส่งผลให้การทำงานมีความรวดเร็วมากยิ่งขึ้น

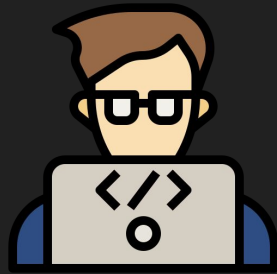


<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

หลักการพื้นฐานของ Git



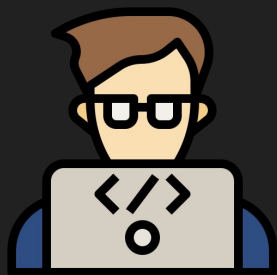
(Local Repository)



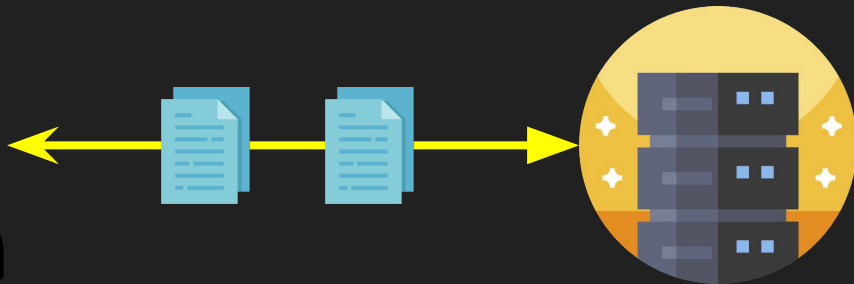
(Remote Repository)

ผู้พัฒนาสามารถแก้ไขโค้ดโปรเจกต์ได้ทุกที่ในรูปแบบ Offline และ Check-In เพื่อจัดเก็บความเปลี่ยนแปลงของ Source Code ลงในฐานข้อมูลภายในเครื่องของตนเอง (Local Repository) โดยไม่จำเป็นต้องติดต่อกับ Git Repository บน Server (Remote Repository)

หลักการพื้นฐานของ Git



(Local Repository)



(Remote Repository)

เมื่อแก้ไขเสร็จแล้วจึงค่อยสั่ง Sync เพื่อให้ Version ฝั่ง Local และ Server อัปเดตเหมือนกันในภายหลังด้วยวิธีการ Pull / Merge / Push



<https://www.youtube.com/c/KongRuksiamOfficial/>

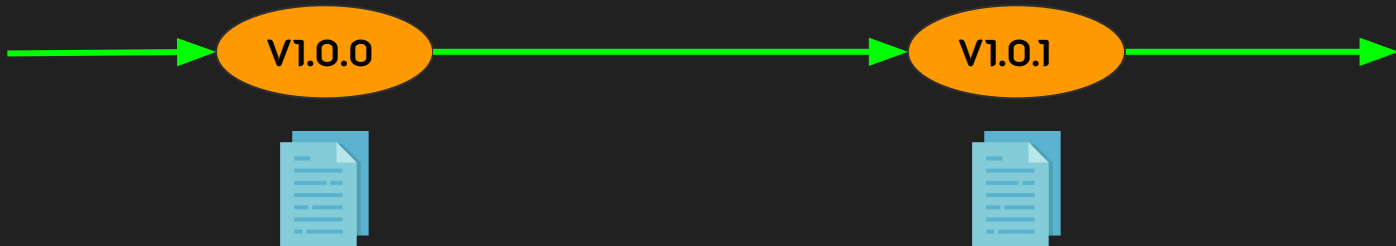


<https://www.facebook.com/KongRuksiamTutorial/>

ทำไมต้องใช้ Git ?



1. ติดตามเวอร์ชันของโค้ดได้ (Track Version)



เมื่อจัดเก็บไฟล์เข้าไปในระบบของ Git จะเรียกว่า Git Repository ซึ่งเก็บสำรองข้อมูลและการเปลี่ยนแปลงของ Source Code ทำให้สามารถย้อนกลับไปดูรายละเอียดการเปลี่ยนแปลงของแต่ละเวอร์ชัน ก่อนหน้าได้

ทำไมต้องใช้ Git ?



2. เครื่องมืออำนวยความสะดวกสำหรับการทำงานเป็นทีม

Git สามารถเก็บบันทึกการเปลี่ยนแปลงของ Source Code เวอร์ชันล่าสุดไว้ที่ **Local Repository** ซึ่งไม่จำเป็นต้องต่ออินเทอร์เน็ตเวลาทำงาน ถ้าต้องการอัปเดตการเปลี่ยนแปลงของ Source Code เวอร์ชันล่าสุดให้กับเพื่อนร่วมทีมก็สามารถที่จะ Push ขึ้นไปเก็บที่ **Remote Repository (Git Hosting : GitHub)** และเพื่อนร่วมทีมก็สามารถดึง (Pull) เวอร์ชันล่าสุดนั้นมารวมที่เครื่องของตนเองได้ ทำให้ Source Code ที่พัฒนาร่วมกันในทีมเป็นเวอร์ชันเดียวกัน

ทำความรู้จักกับ GitHub



เว็บเซิร์ฟเวอร์ที่ให้บริการในการ
ฝากไฟล์ Git หรือ Git ที่ทำงานบน
เว็บไซต์

ทำให้สามารถใช้ Git ร่วมกับคน
อื่นได้ผ่านเว็บไซต์ซึ่งจะมักนิยมใช้
เก็บ Project Open Source ต่างๆ

ทำความรู้จักกับ GitHub



BREAK!



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



ดาวน์โหลดและติดตั้ง Git



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



สมัครใช้งาน GitHub



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



วงจรการทำงานของ Git (Git WorkFlow)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



วงจรการทำงานของ Git

**Working
Directory
(Untracked)**

**Working
Directory
(Tracked)**

Staging Area

**Local
Repository**

**Remote
Repository**



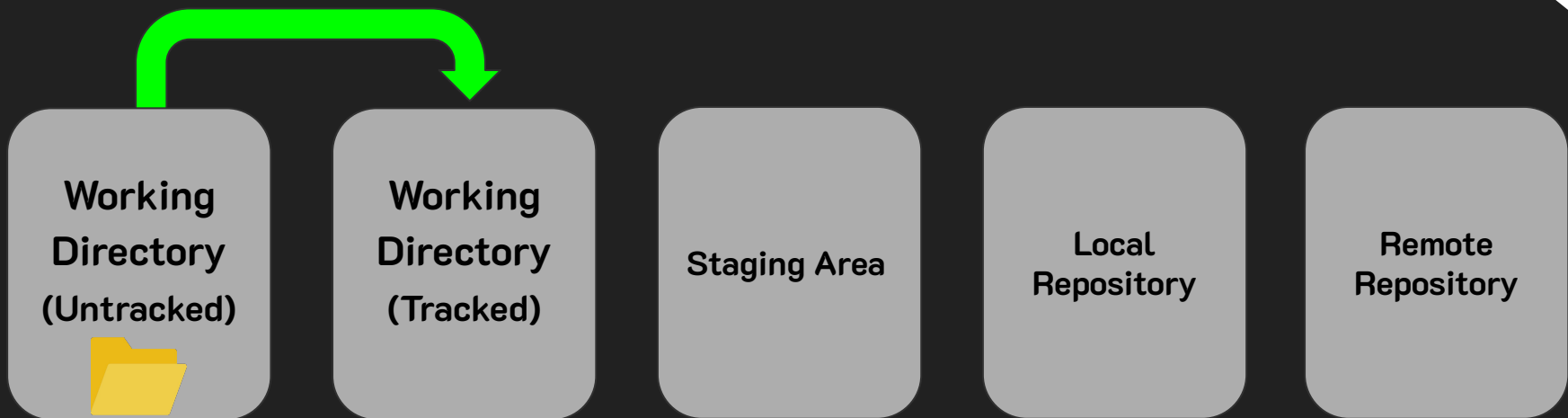
<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



วงจรการทำงานของ Git



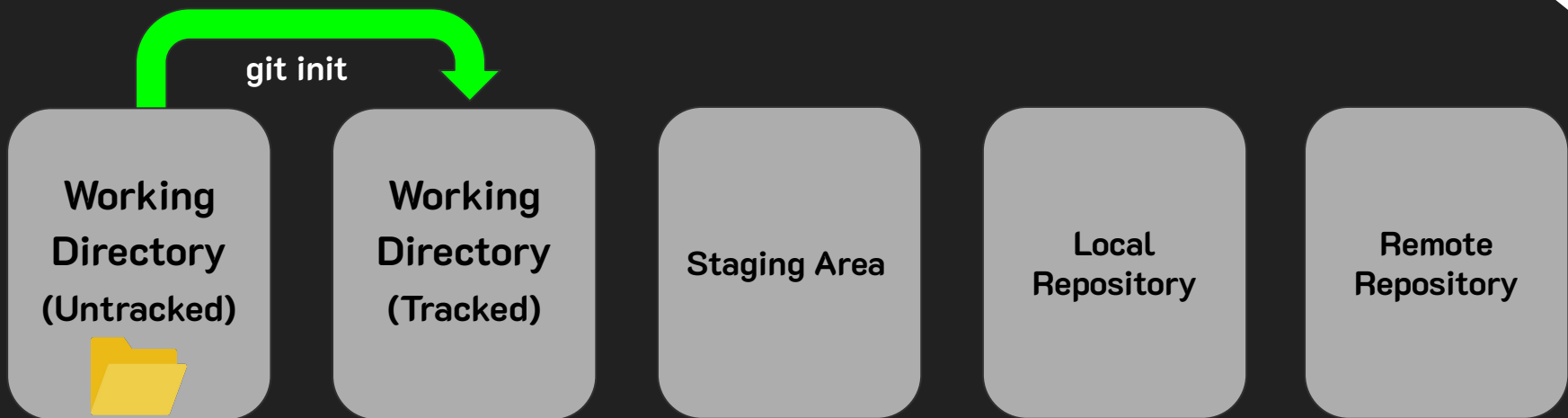
<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>

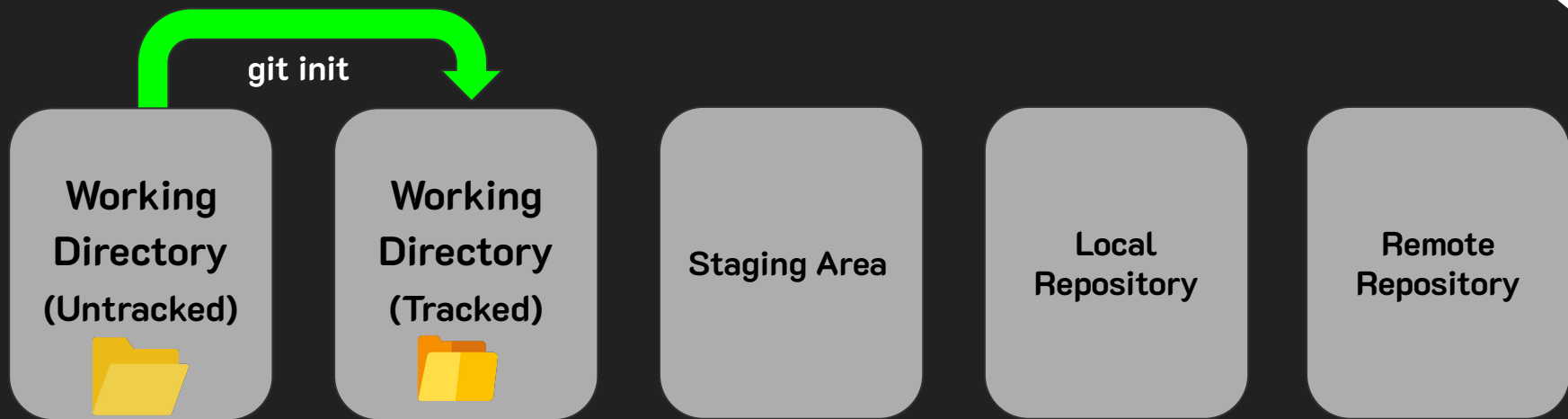


วงจรการทำงานของ Git





วงจรการทำงานของ Git



ใช้ Git มาตรวจสอบ
ไฟล์ในโฟลเดอร์



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>

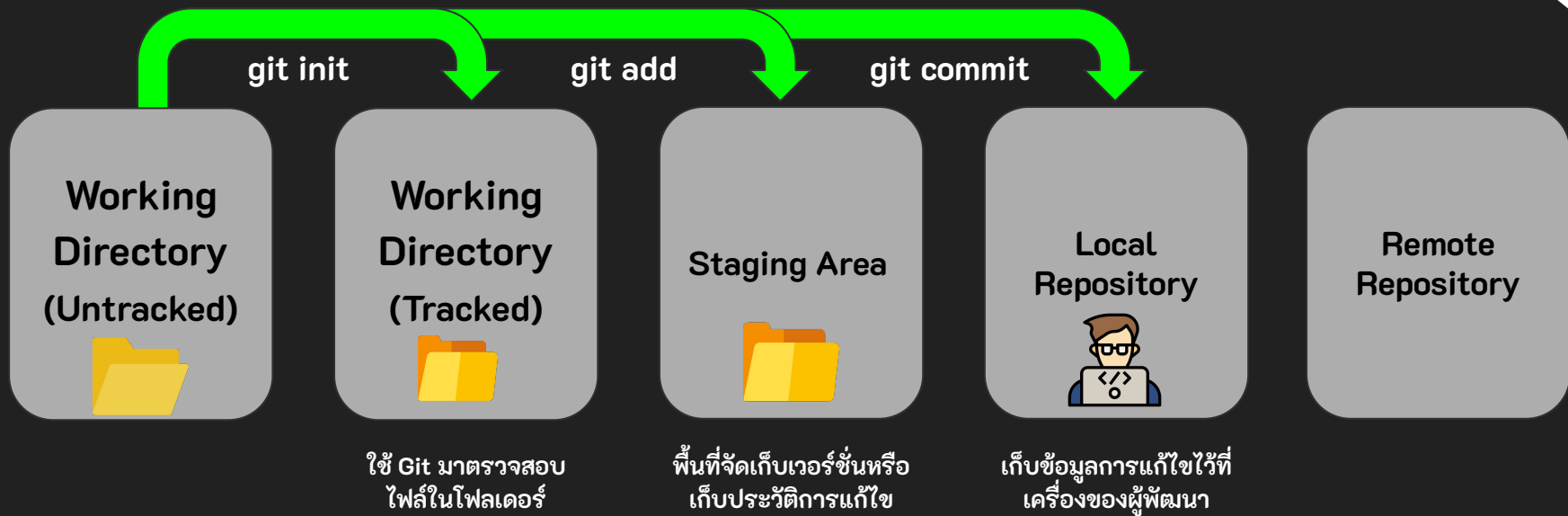


วงจรการทำงานของ Git



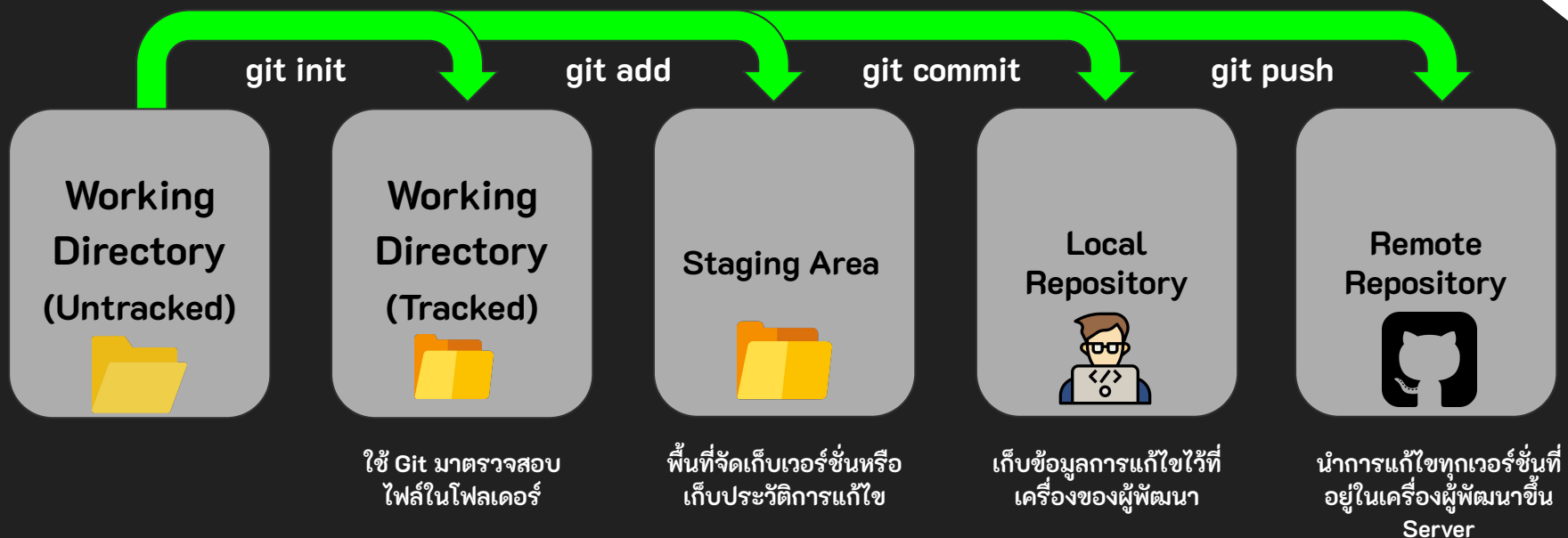


วงจรการทำงานของ Git





วงจรการทำงานของ Git





เริ่มต้นใช้งาน Git เบื้องต้น



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



วงจรการทำงานของ Git





สร้าง Local Git Repository

- git init

การเพิ่มไฟล์เข้าไปใน Staging Area (Check-in)

- git add <file_name> => ระบุไฟล์ เช่น git add index.html
- git add *.html => เพิ่มหลายๆไฟล์พร้อมระบุนามสกุล
- git add . => เพิ่มทุกไฟล์ที่อยู่ภายใต้ Directory ปัจจุบัน





ลบไฟล์หรือโฟลเดอร์ออกจาก Git Repository (Remove)

- `git rm -r --cached .` => ลบทั้งหมด
- `git rm --cached <file_name>` => ระบุไฟล์

ตรวจสอบสถานะ (status) ดูสถานะการเปลี่ยนแปลงของ Repository บนเครื่องของเรา (Local) เช่น การเพิ่ม ,แก้ไข,ลบ ไฟล์ต่างๆ

- `git status`





สถานะการติดตาม (Tracked Status)

- **Modified** หมายถึง มีการแก้ไขไฟล์แล้วแต่ยังไม่เริ่มต้นจัดเก็บลงบน Repository
- **Staged** หมายถึง ได้ทำเครื่องหมาย File ที่ได้ถูกแก้ไขเพื่อบันทึกในเวอร์ชันหน้า
- **Committed** หมายถึง ข้อมูลถูกบันทึกลงใน Repository เรียบร้อยแล้ว





Git Commit , Git Log



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



วงจรการทำงานของ Git





วงจรการทำงานของ Git





เก็บประวัติการแก้ไขถาวร (Commit)

- `git commit -m "Log Message"`

Option

- `m` เป็นพารามิเตอร์สำหรับใส่ข้อความช่วยจำ (Log Message) เพื่ออธิบายการ commit แต่ละเวอร์ชัน
- เมื่อ Commit ไปแล้ว จะได้ SHA-1 Hash เป็น Commit ID (รหัสประจำเวอร์ชัน)
(SHA-1 Hash = 40 ตัวอักษร แต่ตอนอ้างอิงใช้แค่ 7 ตัวอักษรแรก)





ดูประวัติการ Commit (log)

โดยจะแสดง Commit ID, Message, ชื่อผู้เขียน , อีเมล , และเวลาที่ Commit

- git log
- git log --oneline -> แสดงแต่ละLog เหลือบรรทัดเดียว
- git log --graph -> แสดงเป็นเส้น Branch ให้ดูง่ายขึ้น



BREAK!



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



เปรียบเทียบเวอร์ชัน (diff)

ใช้สำหรับตรวจสอบและเปรียบเทียบไฟล์เพื่อดูว่ามีอะไรเปลี่ยนแปลง และแตกต่างไปจากเดิมบ้างเมื่อเทียบกับ Commit ที่ผ่านมา

- `git diff <commit_id>` // แบบระบุ Commit ID
- `git diff <commit_id> <commit_id>` // เปรียบเทียบระหว่างสอง Commit





เปรียบเทียบเวอร์ชัน (diff)

สีสถานะ

- แสดงเครื่องหมาย - และตัวอักษรสีแดงในบรรทัดเดิมก่อนถูกแก้ไขและถูกลบ
- แสดงเครื่องหมาย + และตัวอักษรสีเขียวและโค้ดใหม่ที่ถูกแก้ไขและเพิ่มใหม่





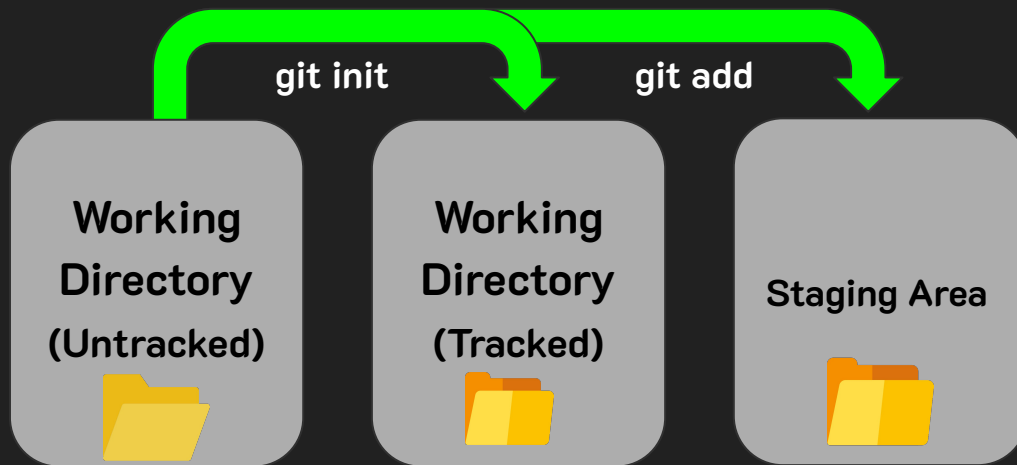
ยกเลิกการแก้ไขไฟล์ (Check-Out)

คำสั่งย้อนกลับไปยัง Commit ล่าสุด หรือยกเลิกการแก้ไขไฟล์

- `git checkout <file-name>`



Git Reset



เป็นการย้อนเวอร์ชันให้กลับไปอยู่ในสภาพก่อนที่จะ add ไฟล์เข้าสู่ Staging Area ซึ่งบางครั้งมีการเพิ่มไฟล์ลงใน Staging Area โดยไม่ตั้งใจ สามารถเอาออกได้ โดยใช้ git reset



BREAK!



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Reset (สำหรับย้อนคืนเวอร์ชัน)

Commit #1

Commit #2

Commit #3

Commit #4



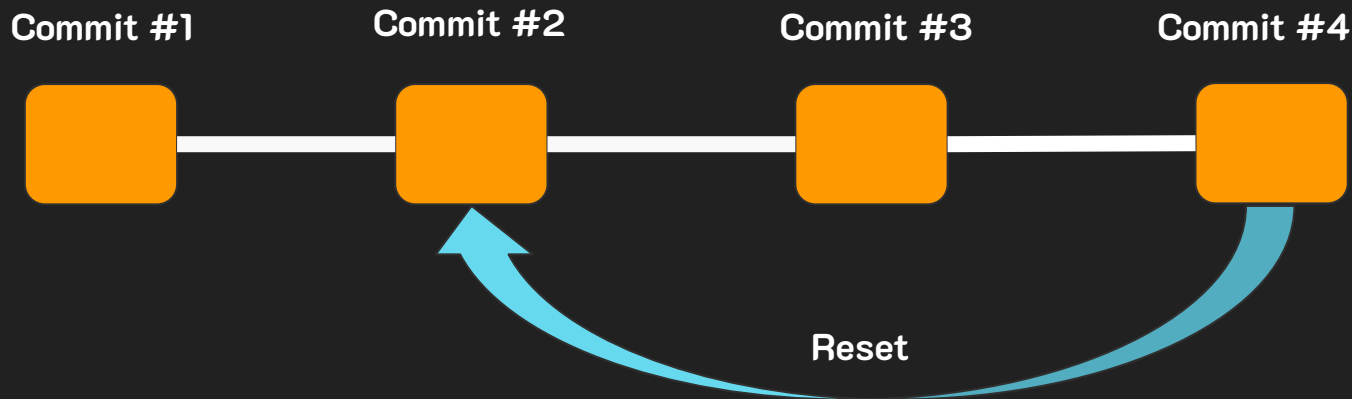
<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Reset (สำหรับย้อนคืนเวอร์ชัน)





Git Reset (สำหรับย้อนคืนเวอร์ชัน)

`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 Branching



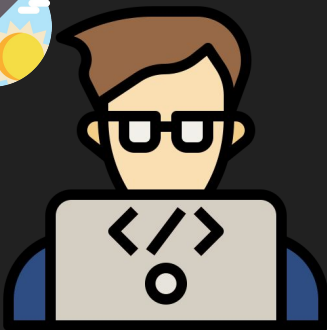
<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

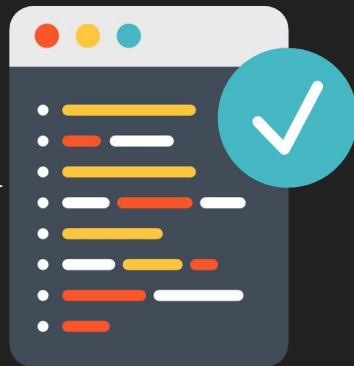


Git Branching



เด็กชายก้อง

เขียนโค้ด



project



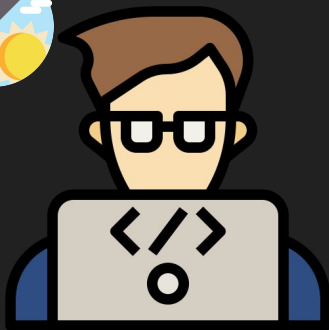
<https://www.youtube.com/c/kongruksiamofficial>



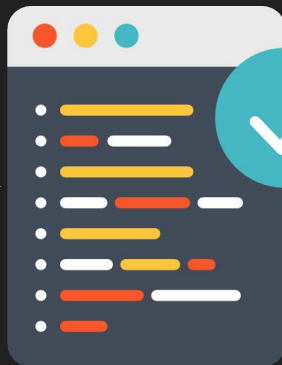
<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching



เด็กชายก้อง



project

- เก็บข้อมูลการขายสินค้า



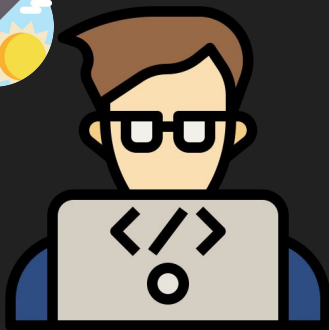
<https://www.youtube.com/c/kongruksiamofficial>



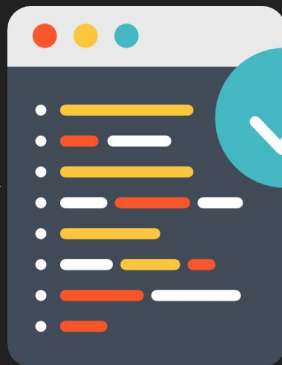
<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

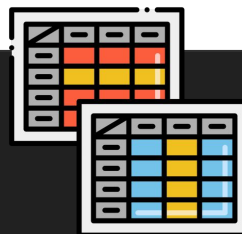


เด็กชายก้อง



project

- เก็บข้อมูลการขายสินค้า



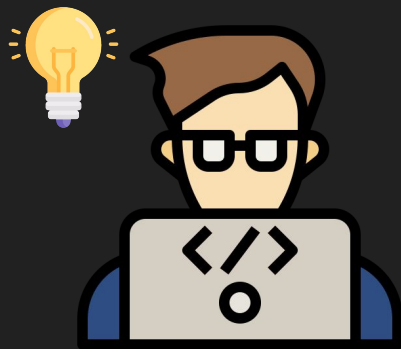
<https://www.youtube.com/c/kongruksiamofficial>



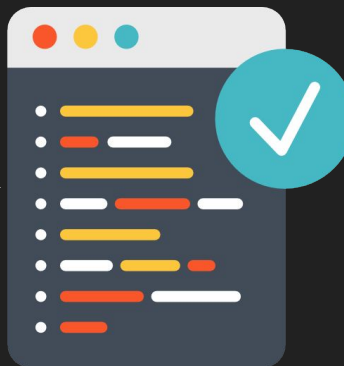
<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

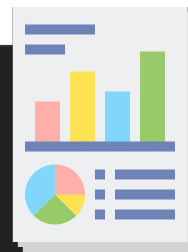


เด็กชายก้อง



project

- เก็บข้อมูลการขายสินค้า
- ****รายงานยอดขายแต่ละเดือน**



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

Commit #1

Commit #2

Commit #3



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

Commit #1

Commit #2

Commit #3



เก็บข้อมูลการขายสินค้า



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

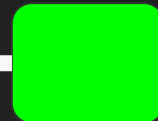
Commit #1

Commit #2

Commit #3

Commit #4

Commit #5



เก็บข้อมูลการขายสินค้า

รายงานยอดขายแต่ละเดือน



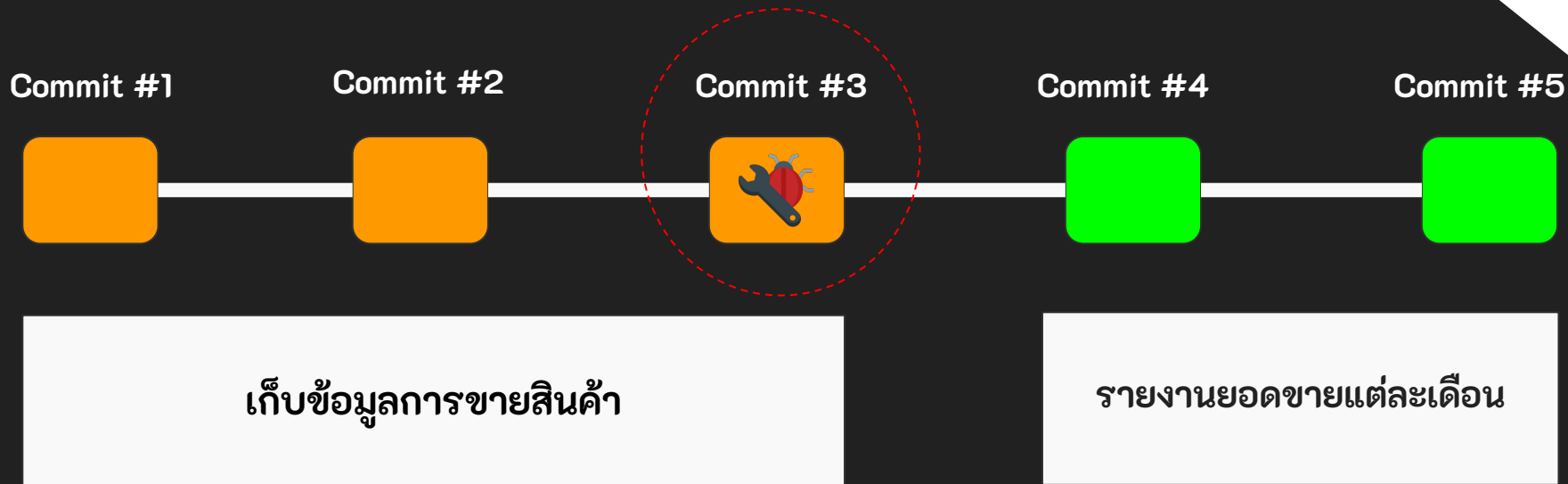
<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching



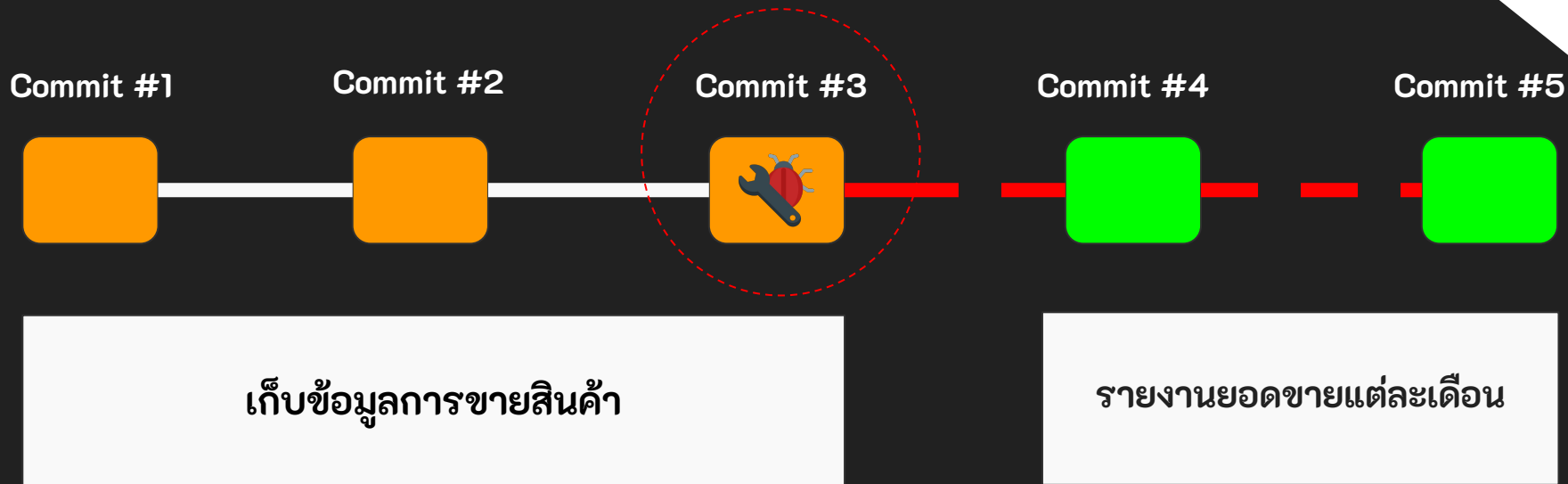
<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>

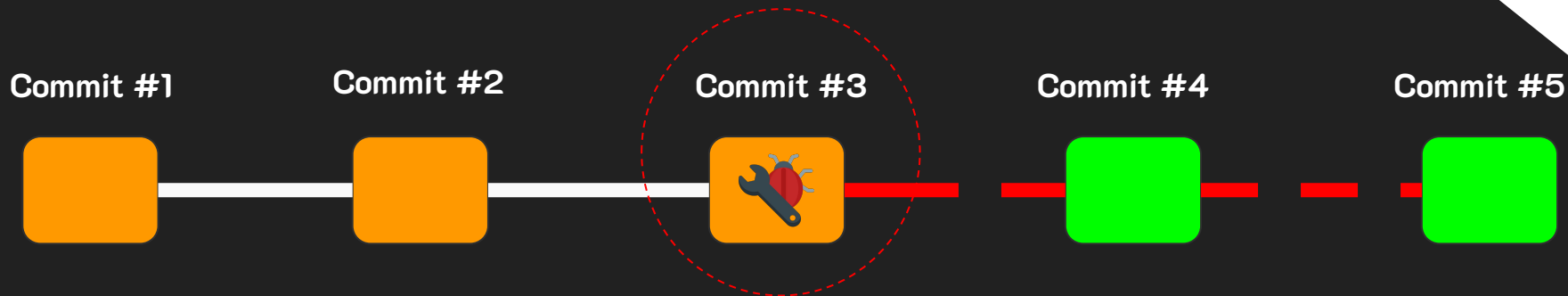


Git Branching





Git Branching



เก็บข้อมูลการขายสินค้า

Git Reset



Git Branching

Commit #1

Commit #2

Commit #3



เก็บข้อมูลการขายสินค้า



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

Commit #1

Commit #2

Commit #3



เก็บข้อมูลการขายสินค้า



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

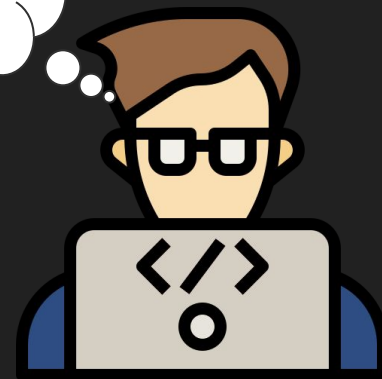
Commit #1

Commit #2

Commit #3



เก็บข้อมูลการขายสินค้า



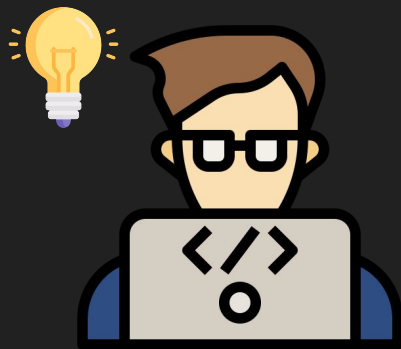
<https://www.youtube.com/c/kongruksiamofficial>



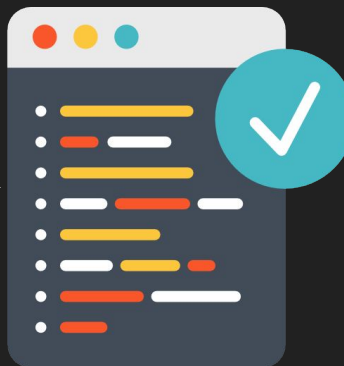
<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching



เด็กชายก้อง



project

ทำยังไงจึงจะสามารถพัฒนา
โปรเจกต์ใหม่ / เพิ่มฟีเจอร์เข้าไป
ในโปรเจกต์เก่าและใช้งานระบบใน
เวอร์ชันเก่าได้ด้วย ?



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

Commit #1

Commit #2

Commit #3



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

Commit #1

Commit #2

Commit #3



Master เป็น Branch Default (กิ่งหลัก)

ตอนใช้คำสั่ง git init



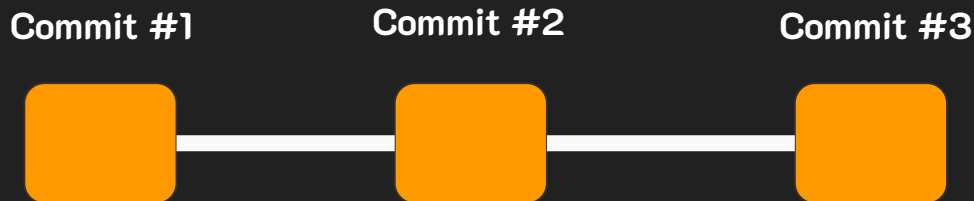
<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching



Master เป็น Branch Default (กิ่งหลัก)
ตอนใช้คำสั่ง git init



Master Branch



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>



Git Branching

Commit #1

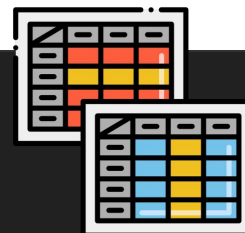
Commit #2

Commit #3



Master Branch

- เก็บข้อมูลการขายสินค้า



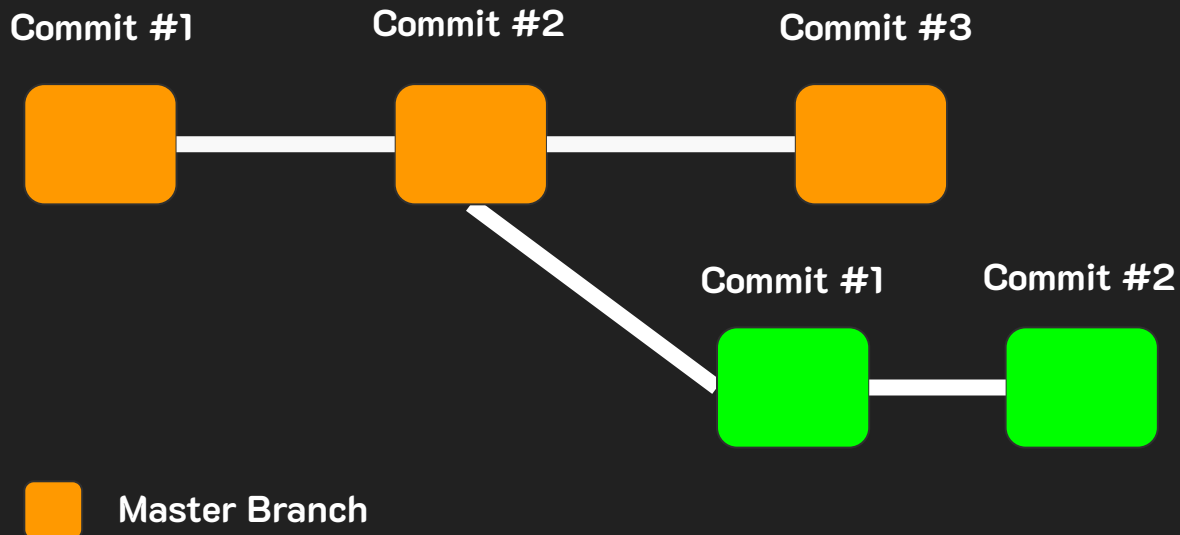
<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>

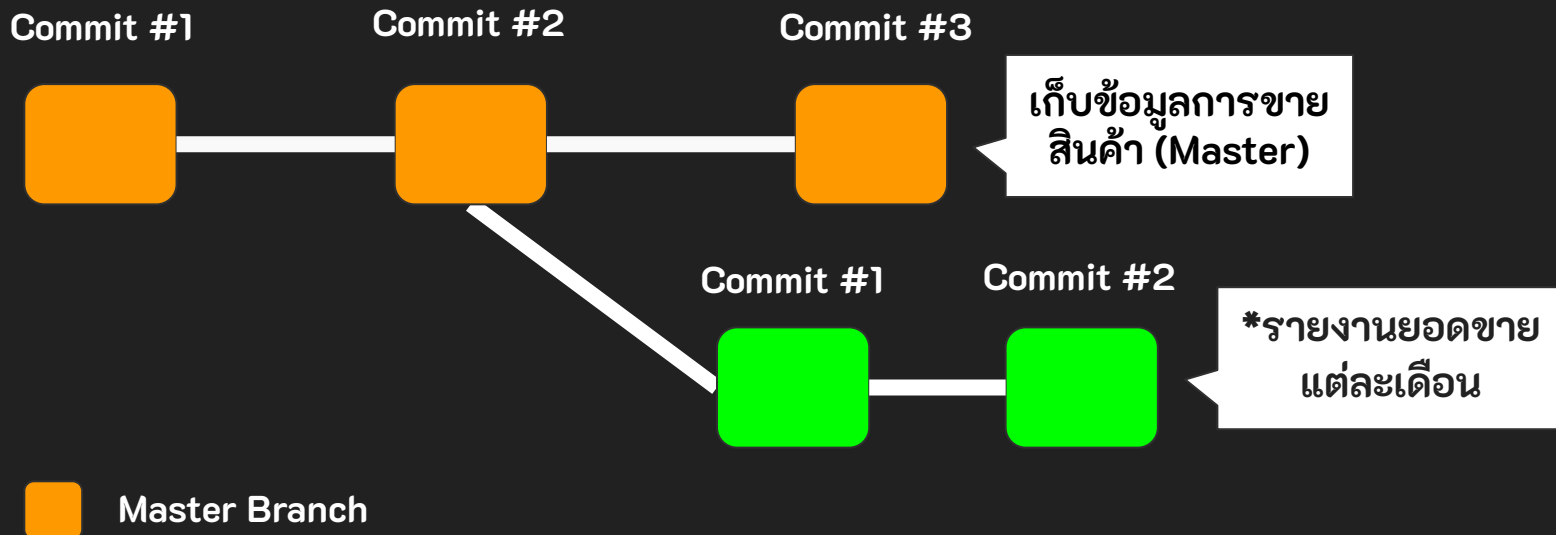


Git Branching



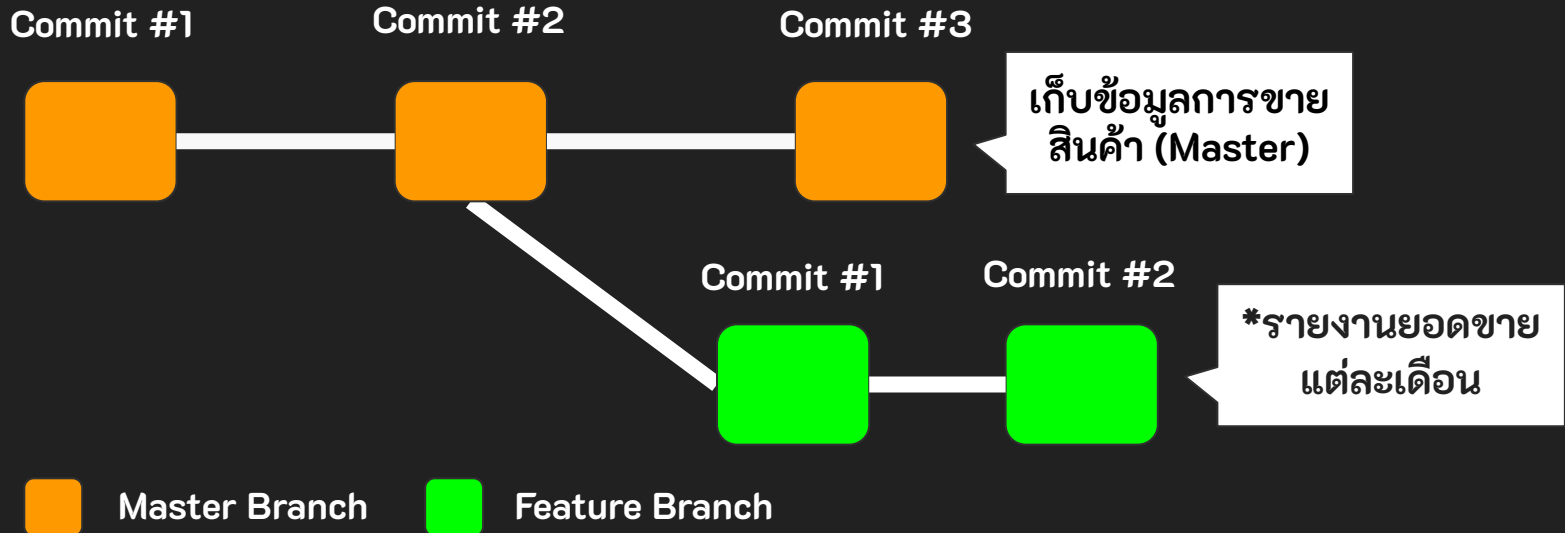


Git Branching



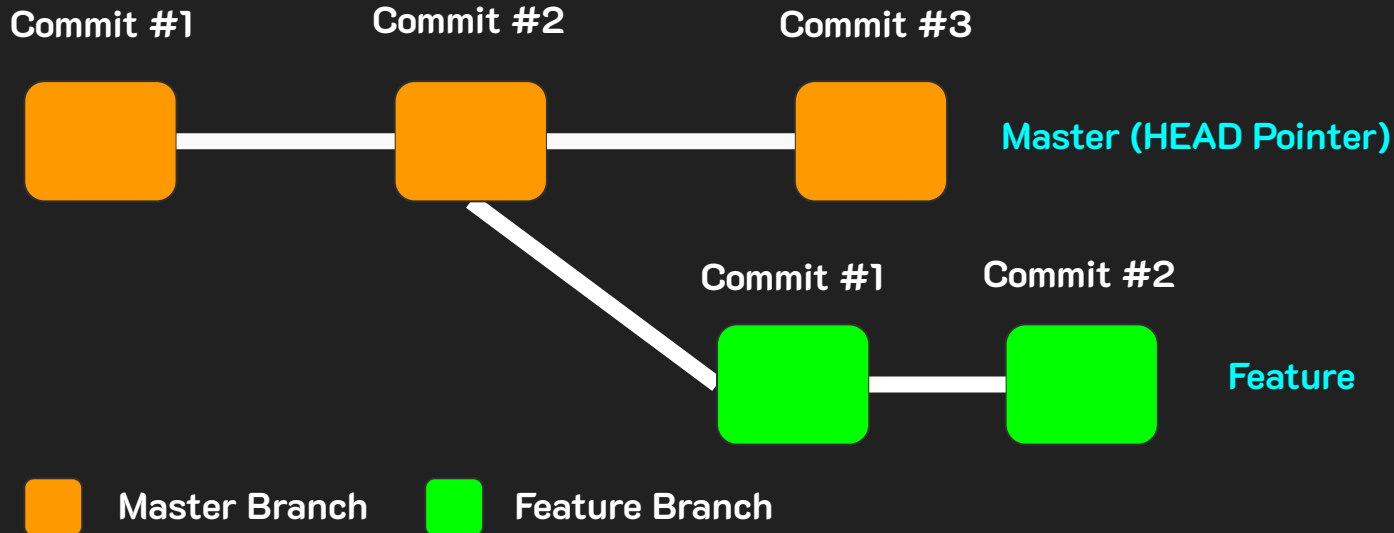


Git Branching



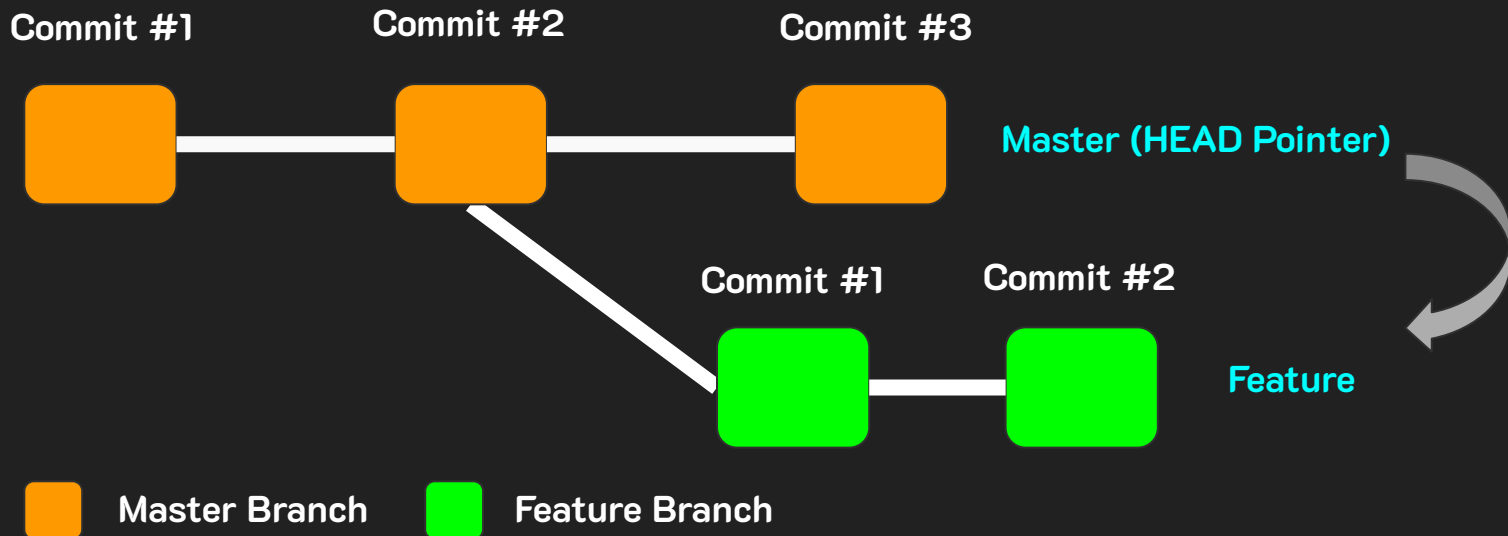


Git Branching



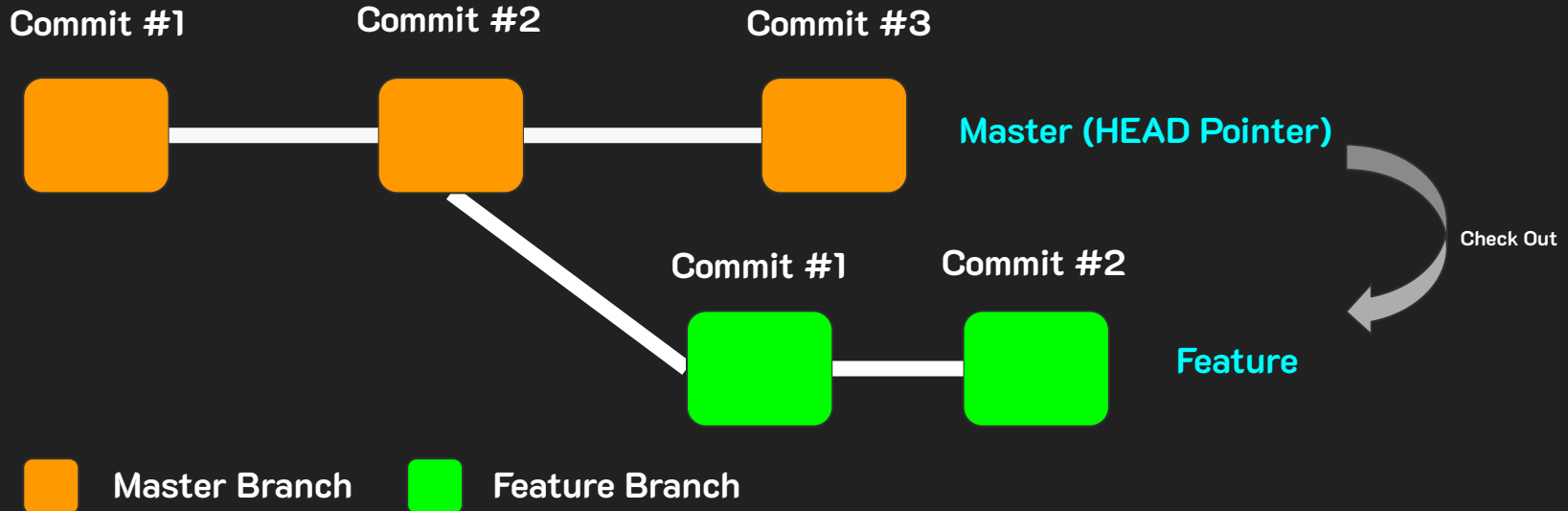


Git Branching



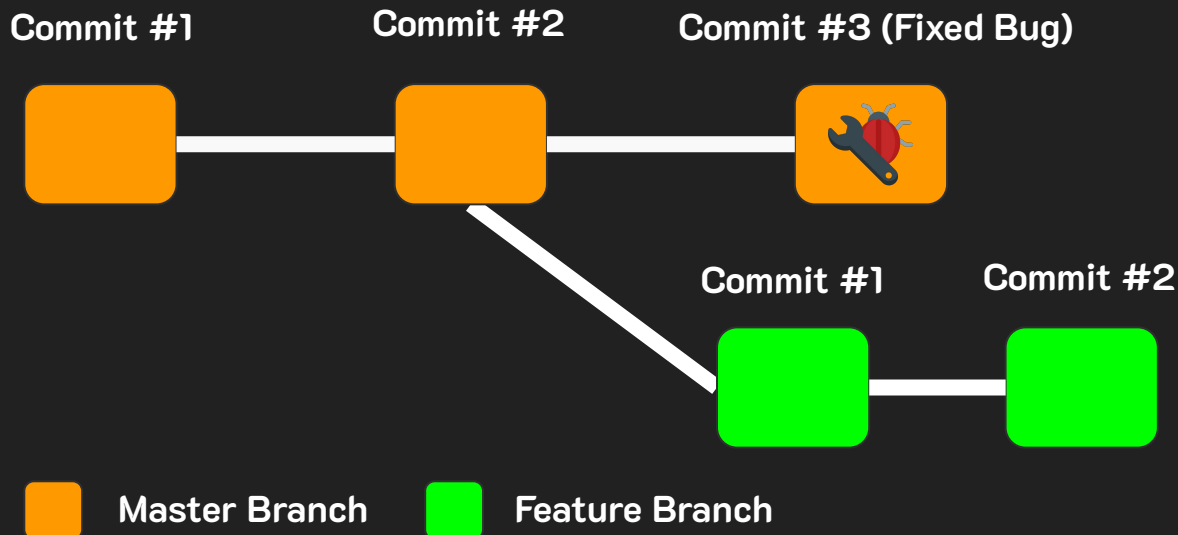


Git Branching



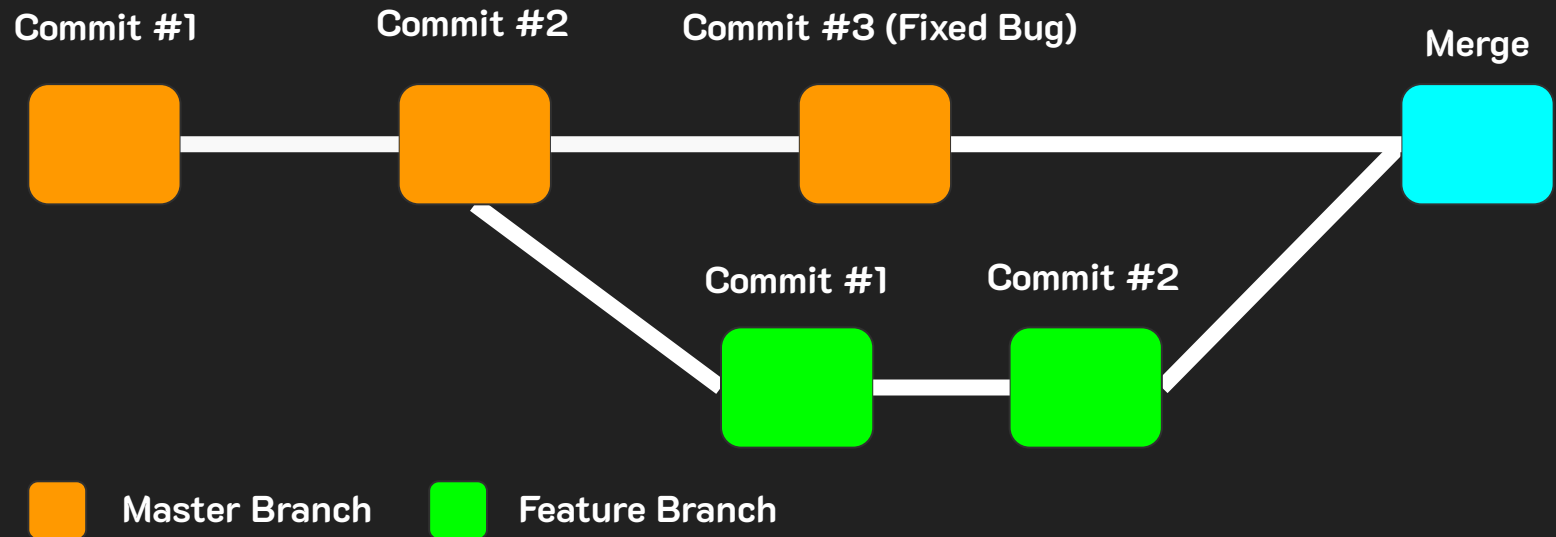


Git Branching





Git Branching



BREAK!



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



การจัดการ Git Branch เบื้องต้น



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



คำสั่งที่เกี่ยวข้องกับ Git Branching

- git branch
- git checkout
- git merge





คำสั่งที่เกี่ยวข้องกับ Git Branching

- การแสดงชื่อ Branch

`git branch`

- การสลับและสร้าง Branch

`git checkout -b <ชื่อ branch>` (ห้ามตั้งชื่อเว้นวรรค)





คำสั่งที่เกี่ยวข้องกับ Git Branching

- การลบ Branch

`git branch -d <ชื่อ branch>`

- สลับไป Branch ที่ต้องการ

`git checkout <ชื่อ branch>`





คำสั่งที่เกี่ยวข้องกับ Git Branching

- สลับไป Branch หลัก

`git checkout master`

- การรวม Branch

`git merge <ชื่อ branch>`



BREAK!



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Remote Repository (Push / Pull)



<https://www.youtube.com/c/KongRuksiamOfficial/>

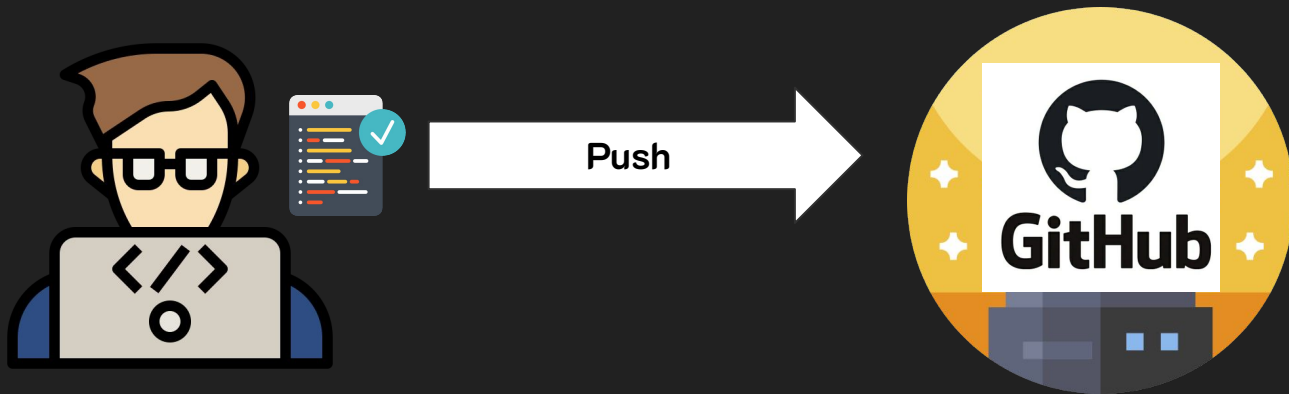


<https://www.facebook.com/KongRuksiamTutorial/>



Git Push (ผลัก)

เป็นคำสั่งที่ใช้สำหรับนำสิ่งที่อยู่ในเครื่องของเรา (Local Repository)
ไปอัปเดตให้กับ Remote Repository (Server)



<https://www.youtube.com/c/kongruksiamofficial>

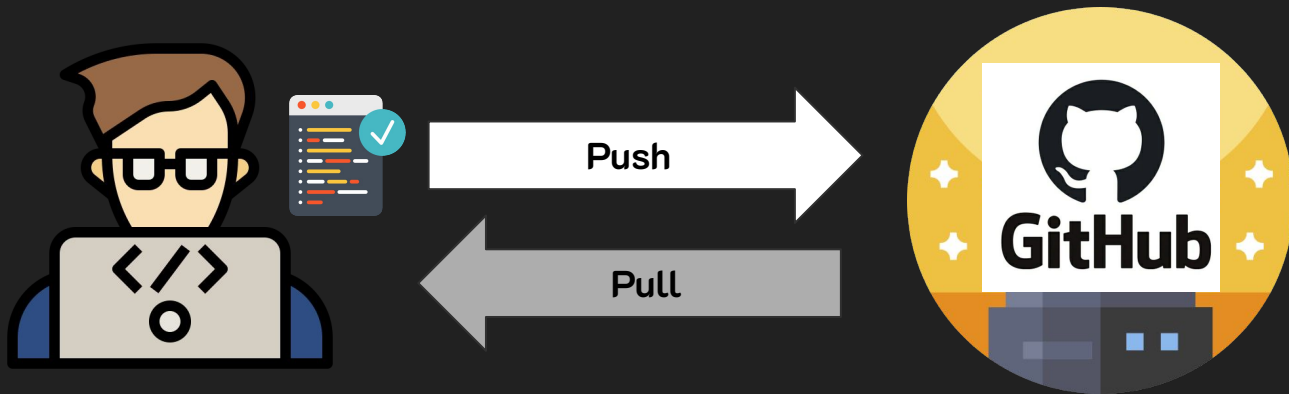


<https://www.facebook.com/KongRuksiamTutorial/>



Git Pull (ดึง + รวม)

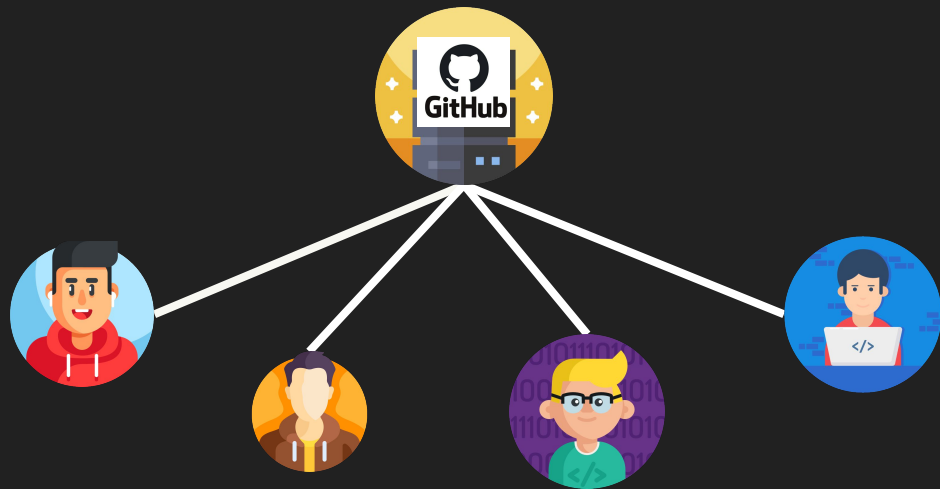
เป็นคำสั่งที่ใช้สำหรับนำสิ่งที่อยู่บน Remote Repository (Server)
มาอัปเดตในเครื่องของเรา (Local Repository)





Git Clone

เป็นคำสั่งที่ใช้สำหรับนำเอาโปรเจกต์ที่อยู่ใน Remote Repository (Server) มาไว้ในเครื่องของเราหรือคนในทีม (Local Repository)



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>

เข้าเรียนเนื้อหาได้ที่



เรียนรู้การใช้งาน Git & GitHub สำหรับผู้เริ่มต้น [2022]






วิดีโอ 21 รายการ • การดู 269 ครั้ง • อัปเดตแล้ววันนี้



KongRuksiam
Official

ติดตามแล้ว



- 1  **Git & GitHub ตอนที่ 1 - รู้จักกับ Version Control**
KongRuksiam Official 7:47
- 2  **Git & GitHub ตอนที่ 2 - สาเหตุที่ต้องใช้ Version Control**
KongRuksiam Official 13:03
- 3  **Git & GitHub ตอนที่ 3 - วัฒนาการของ Version Control**
KongRuksiam Official 16:18
- 4  **Git & GitHub ตอนที่ 4 - รู้จักกับ Git & GitHub**
KongRuksiam Official 9:40
- 5  **Git & GitHub ตอนที่ 5 - สมัครใช้งาน Github**
KongRuksiam Official 6:36



สแกนเพื่อเข้าเรียน



<https://www.youtube.com/c/kongruksiamofficial>



<https://www.facebook.com/KongRuksiamTutorial/>