**11. โปรแกรม Version control มีประโยชน์อย่างไร**

**ตอบ** Version Control มันจะช่วยทำให้เราทำงานกับทีมได้ง่ายขึ้น มันสามารถบอกว่าได้ว่า ตรงนี้ถูกเปลี่ยนแล้วนะ ใครเป็นคนแก้ แก้เมื่อไหร่ แล้วแก้เพื่ออะไร ตัว Source Code ก็จะถูก Sync กับตัว Server ตลอดเวลา ทำให้มั่นใจได้ว่า Source Code ที่เราทำงานอยู่เป็น Version ล่าสุดตลอดเวลา

**12. ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คืออะไร**

**ตอบ** Git เป็น distributed version control system ที่เราสามารถแยกversion control ออกเป็น repo ย่อยๆได้ แต่ละ repo ย่อยก็สามารถทำ versioning ของตัวเองได้อีกที

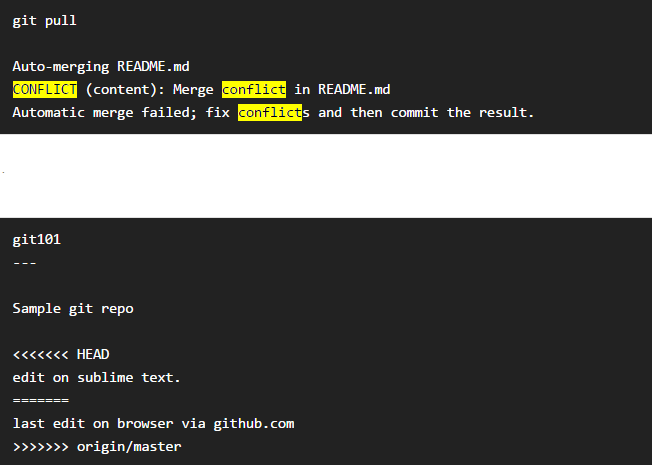
**13.  ข้อได้เปรียนของ centraized version control เมื่อเทียบกับdistributed version control คืออะไร**

**ตอบ** ส่วน SVN นั้นเป็น centralized version control system คือ vesion ทั้งหมดจะถูกเก็บที่ remote repo

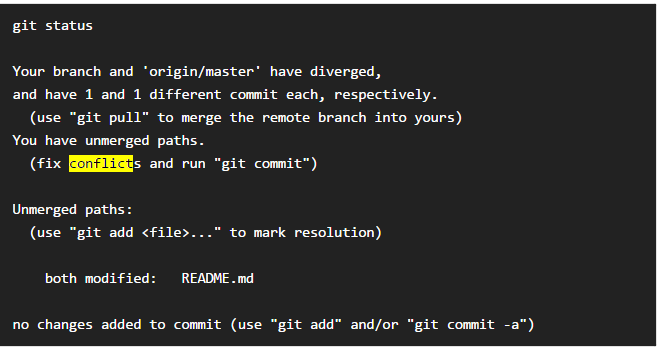
**14.บอกแนวทางในการแก้ไข conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน**

**ตอบ** เกิดกรณีที่มีการแก้ไขไฟล์เดียวกัน เวลาเดียวกัน Git จะไม่รู้ว่าเราควรจะใช้ไฟล์ไหนดี วิธีการแก้ คือ ทำการ edit แล้ว commit ไปใหม่

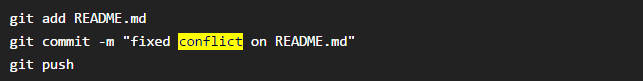
ตัวอย่างของ file ที่เกิด conflict



วิธีแก้ลบโค้ดส่วนเกินออก แล้วแก้ไขใหม่ให้เรียบร้อย จากนั้นเช็คสถานะ



ก็จะ commite และ push ได้ตามปกติ



**15.บอกแนวทางในการลด conflict ที่จะเกิดขึ้นจากการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน**

**ตอบ**  จะต้องทำการ merge การเปลี่ยนแปลงเข้ามายัง master branch หลังจากที่พัฒนาเสร็จแล้วเสมอ เพราะเราจะสามารถเปลี่ยนแปลง แก้ไข และตรวจสอบได้

**16.Git คืออะไรแตกต่างจาก Github อย่างไร**

**ตอบ**Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม

Git แตกต่างจาก Github คือ Github เป็นเว็บเซิฟเวอร์ที่ให้บริการในการฝากไฟล์ Git (ทั่วโลกมักนิยมใช้ในการเก็บโปรเจ็ค Open Source ต่างๆ ที่ดังๆ ไม่ว่าจะเป็น Bootstrap, Rails, Node.js, Angular เป็นต้น)

**17.จุดประสงค์หลักในการ branch คืออะไร**

**ตอบ** คือการรวบรวมไฟล์ที่เป็นระบบงานที่อยู่ฝน master branch จะต้องสามารถ deploy ได้เสมอ เหมาะสำหรับการทำงานเป็นทีม คนในทีมจะต้องสามารถดึง code จาก master branch ไปทำการ build และ deploy ได้ทันที โดยไม่ error  สมมติ ในบริษัทมี 2 ทีม ประชุมกันแล้วว่าจะให้ 2 ทีมนี้ทำกันคนละฟีเจอร์ ก็แตก branch ไปให้ทีมละ branch โค้ดก็จะได้ไปปะปนกัน ในขั้นตอนการพัฒนา (พัฒนาเสร็จค่อยเอามันมารวมกัน) git นั้นมันไม่ได้บอกในตัวมันเองหรอกว่า ต้องแตก branch อย่างนั้นอย่างนี้ เท่านั้น branch เท่านี้ branch ไม่เลย ต้องไปแตกกันเอาเอง ทีมนี้มีคุณสมบัติแบบนี้ทีมนั้นมีคุณสมบัติแบบนั้น  ไปประชุมจัดการกันซะ

**18.Fast forward merge คืออะไรและทำไมการ push ไปที่ remote repo จึงควรจะต้อง merge แบบนี้**

**ตอบ** Fast-forward ซึ่งหมายถึงหมายเลข commit ของ branch master ย้ายไปใช้ตัวเดียวกับของ hi-name เท่านั้นเอง ยังไม่ใช่การ merge ที่แท้จริงแต่อย่างใด

**19.หน้าที่หลักของคำสั่ง git pull คืออะไร**

**ตอบ** ใช้ดึงความเปลี่ยนแปลงจาก remote มายัง local และรวมเข้าด้วยกัน (มีค่าเท่ากับfetch+merge)

**20.แผนภาพนี้ต้องการสื่อความหมายอะไร**

**ตอบ** workflow มันคือแนวทางที่จะทำให้การทำงานมันราบรื่นนั่นแหละ แต่มันมีแนวทางแนวทางหนึ่งที่เป็นที่นิยม เขาตั้งชื่อมันว่า gitflow เมื่อจะพัฒนาฟีเจอร์ใหม่ ให้แตก branch Feature มาจาก Develop ให้ทีมพัฒนาโค้ดกันใน branch เมื่อพัฒนาเสร็จแล้วให้ merge โค้ดเข้า develop