

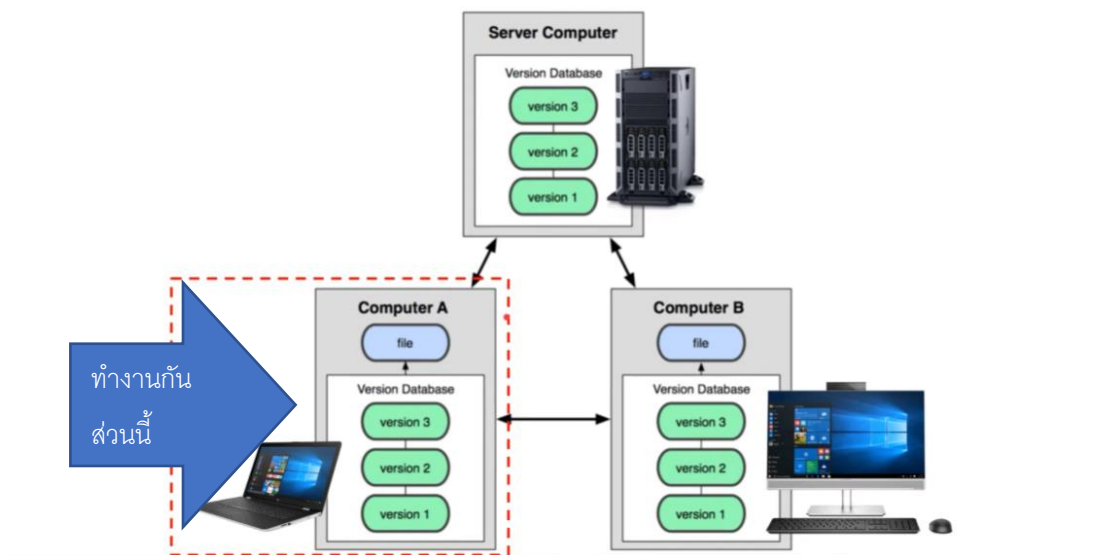
การใช้งาน Git ด้วยคำสั่ง command line

ในบทนี้เราจะมาลองใช้ Git โดยไม่ผ่าน Git Desktop กัน เนื่องจากเวลาเราทำงานกันจริงๆ เราสามารถใช้ VS code ในการทำงานคู่กับ Github ได้เลย โดยไม่ต้องเสียเวลาผ่านโปรแกรมตัวที่ 3 แต่ความยากของมันก็คือ เราจะต้องจำคำสั่งในการใช้งาน Git ให้ได้

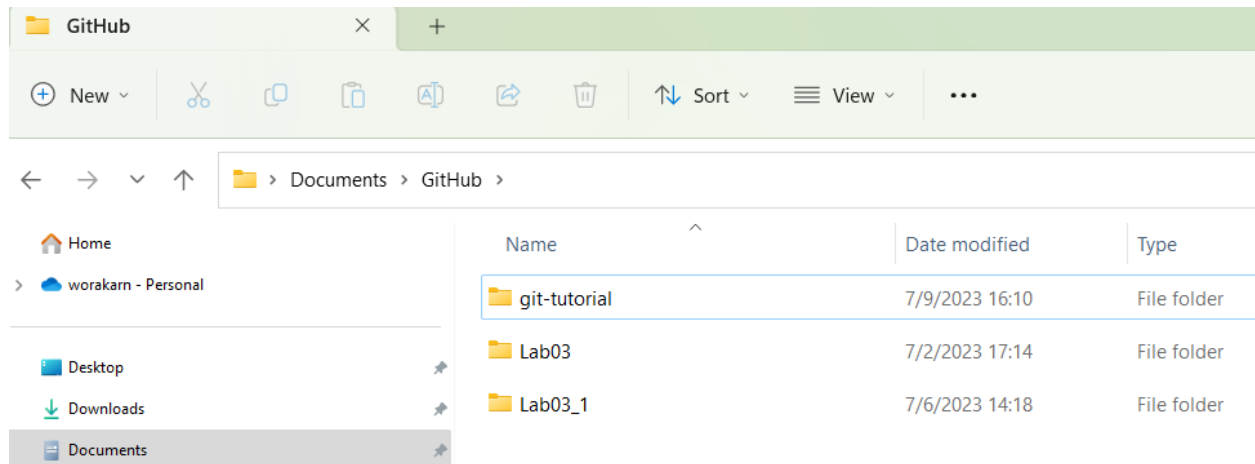
Local Repository

เวลาที่เรารเริ่มสร้าง Project นั้นปกติแล้ว เราจะทำในเรื่องเราก่อน แล้วค่อย Upload ไปที่ Github ตั้งใน นั้น Lab ครั้งนี้เราจะสร้างไฟล์ต่างๆ ไว้ในเรื่องเราก่อนจากนั้นก็จะเรียนรู้คำสั่งต่างๆ ของ Git กับส่วนของ Local นะครับ

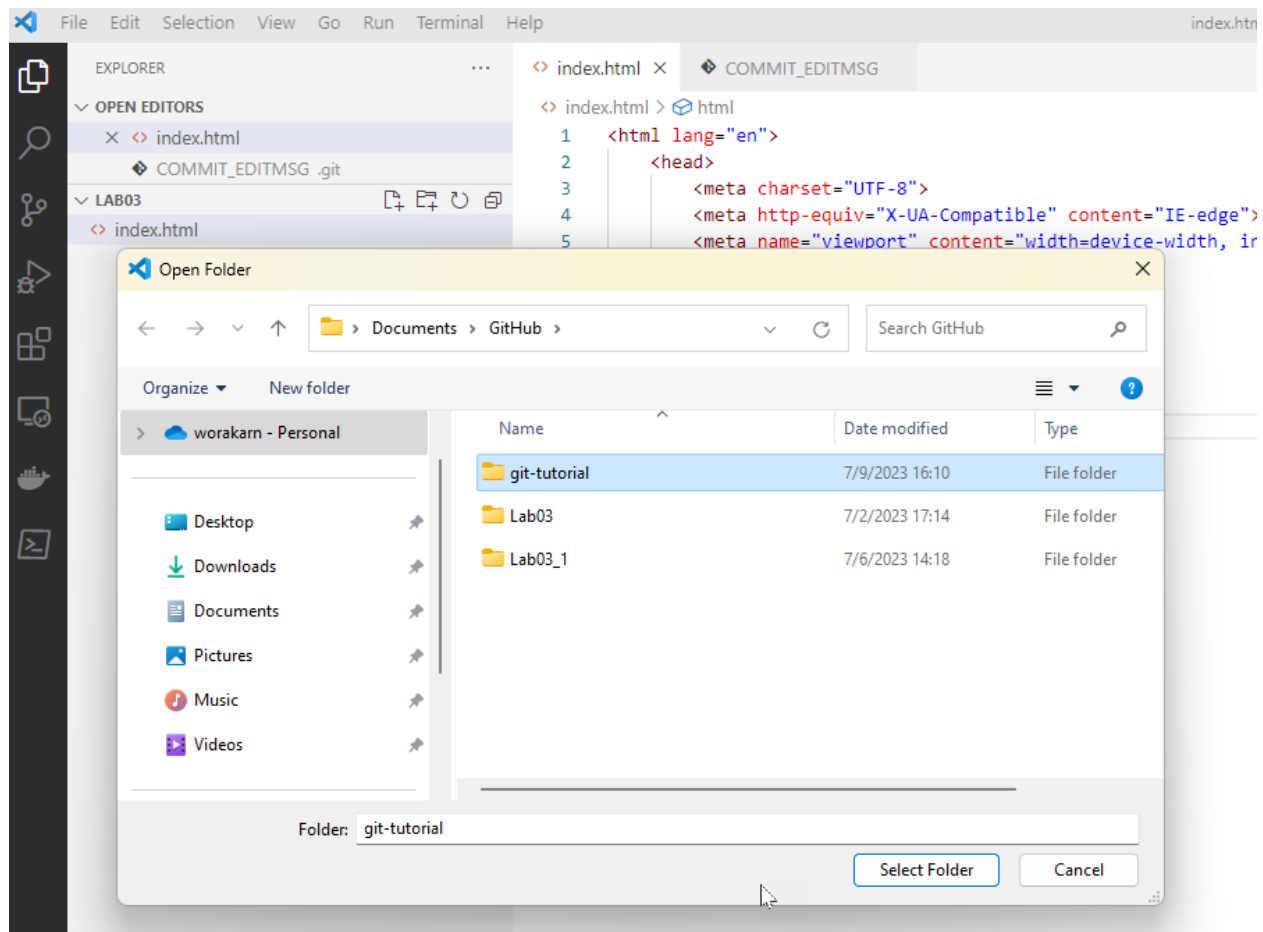
Distributed Version Control Systems



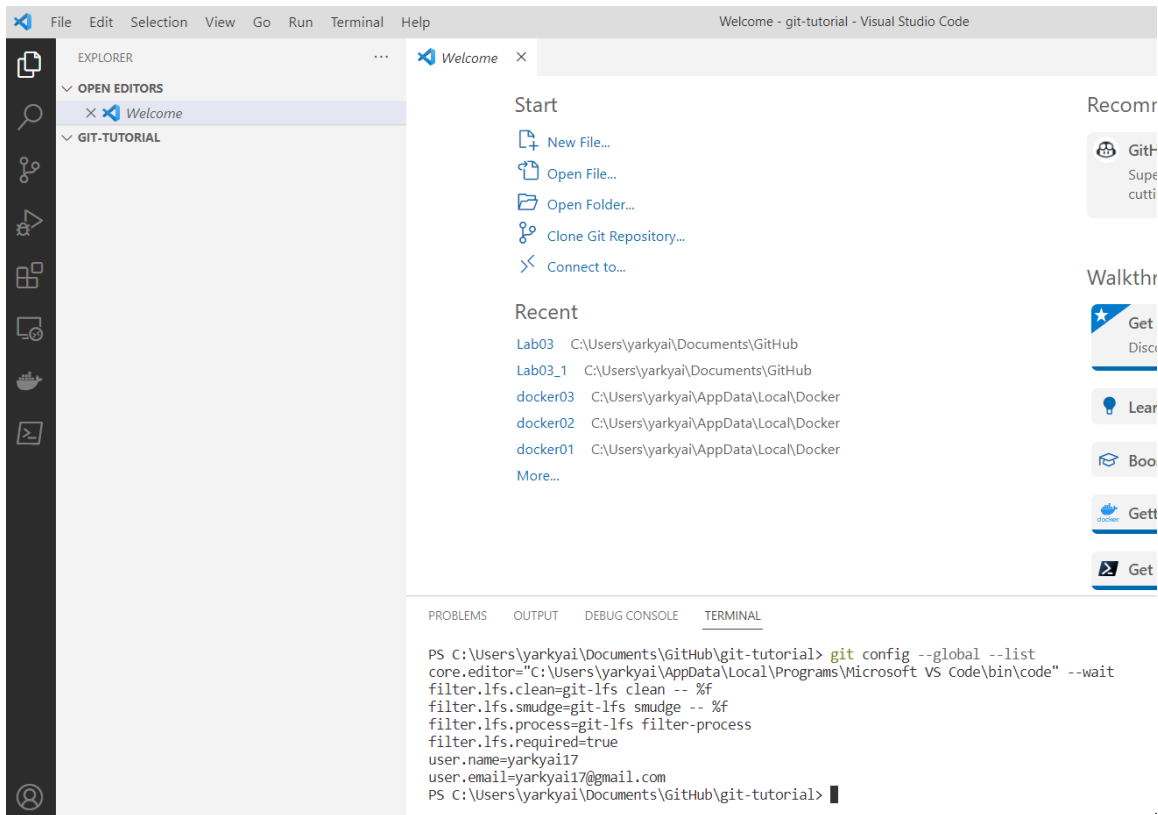
สร้าง folder git-tutorial ที่ root ของ git (ของผมอยู่ที่ C:\Users\yarkyai\Documents\GitHub)



เปิด VS Code เลือก File และ Open folder แล้วเลือก git-tutorial ที่เราได้สร้างเอาไว้



เลือก Terminal แล้ว new terminal ทดสอบพิมพ์คำสั่ง `git config --global --list` ซึ่งเป็นคำสั่งสำหรับแสดงคุณสมบัติของ git ทั้งหมดใน Global



จากรูปจะเห็นว่าโปรแกรมจะแสดง user และ email ที่เราได้ทำไว้เมื่อสัปดาห์ก่อน

ต่อมาเราจะสร้าง git repository ฝั่ง Local ก่อน ด้วยคำสั่ง `git init`

Git Init : สร้าง Git Repository

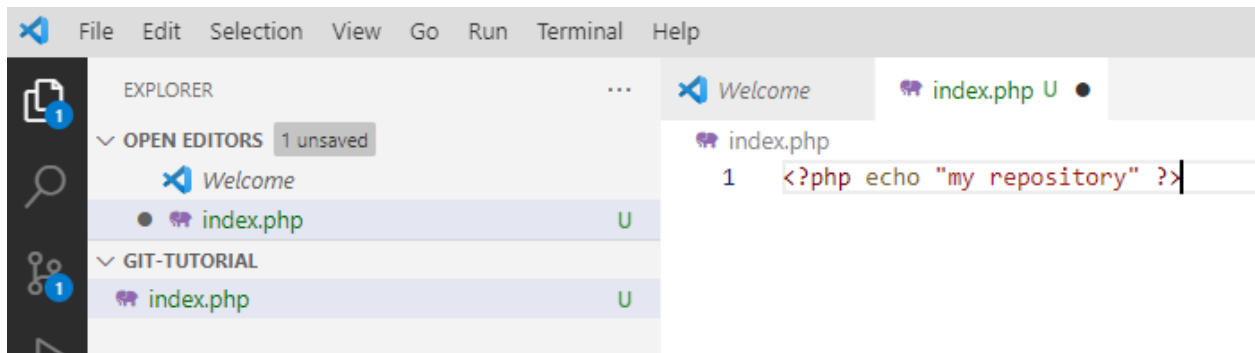
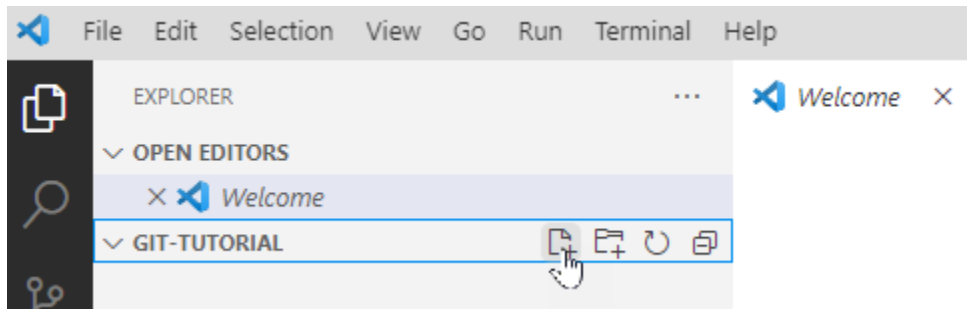
```
git init
```

พิมพ์คำสั่ง `git init` เพื่อบอกเครื่องว่าเราจะให้ folder ดังกล่าวเป็น repository ของเรา

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git config --global --list
core.editor="C:\Users\yarkyai\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
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
user.name=yarkyai17
user.email=yarkyai17@gmail.com
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git init
Initialized empty Git repository in C:/Users/yarkyai/Documents/GitHub/git-tutorial/.git/
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █ █
```

ชื่อว่า index.php และเขียนโปรแกรมตามรูป



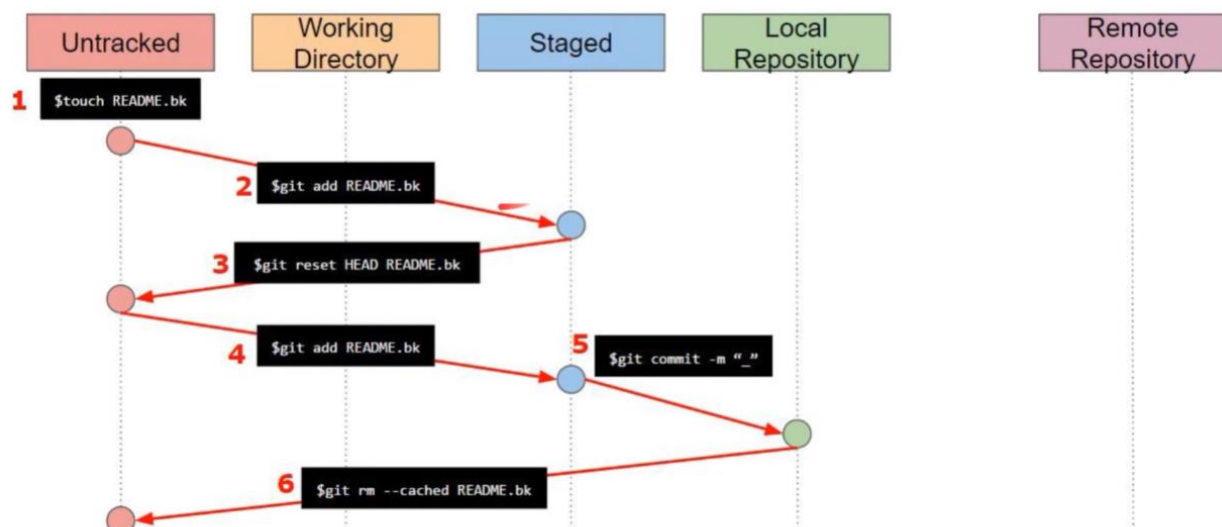
ทำการ save file จะเห็นว่าสถานะของ file จะเป็น U

สถานะของไฟล์เมื่อเราทำงานบน Git

Git Status

- **Untracked** สถานะไฟล์ยังไม่ได้ถูกเก็บไว้ในระบบ Git
- **Working Directory** สถานะที่มีเปลี่ยนแปลงหรือแก้ไขไฟล์
- **Staged** สถานะเตรียม Commit เพื่อยืนยันการเปลี่ยนแปลงก่อนเก็บลงในสถานะ Local Repository
- **Local Repository** สถานะเก็บบันทึกข้อมูลไฟล์ที่เปลี่ยนแปลงลงใน Git Repository แบบ Local (ที่เครื่องตัวเอง)
- **Remote Repository** สถานะเก็บบันทึกข้อมูลไฟล์ที่เปลี่ยนแปลงลงใน Git Repository แบบ Hosting (ที่เครื่องเซิร์ฟเวอร์)

การทำงานของสถานะต่าง



คำสั่งพื้นฐานของ Git

Git Add : เพิ่มไฟล์ใหม่หรือไฟล์ที่แก้ไขไปสู่ Staged

```
git add <file_name>
git add README.md      #เพิ่มไฟล์ชื่อ README.md เข้าไปที่สถานะ Staged
git add .               #ใช้ในกรณีที่มีหลายๆ ไฟล์และต้องการเพิ่มเข้าไปทั้งหมด
```

Git Commit : เก็บไฟล์ Staged เข้าไปใน Local Repository

```
git commit -m "message"    #ยืนยันการเปลี่ยนแปลงพร้อมข้อความ
```

Git Clean : แสดงและลบไฟล์ที่อยู่ในสถานะ Untracked

```
git clean -n              #แสดง Source Code ที่อยู่ในสถานะ Untracked
git clean -df             #ลบ Source Code ที่อยู่ในสถานะ Untracked
```

มาทดลองคำสั่งต่างๆ กัน ตอนนี้ใน VS code เราจะเห็นว่าไฟล์ของเราอยู่ในสถานะ U หรือเราสามารถพิมพ์คำสั่ง git status ใน terminal โปรแกรมก็จะแสดงข้อมูลว่ามีไฟล์ index.php อยู่ในสถานะ untracked

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git init
Initialized empty Git repository in C:/Users/yarkyai/Documents/GitHub/git-tutorial/.git/
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.php

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

ต่อมาเราจะทำการ add file เข้าไปใน git ของเราด้วยคำสั่ง git add filename

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

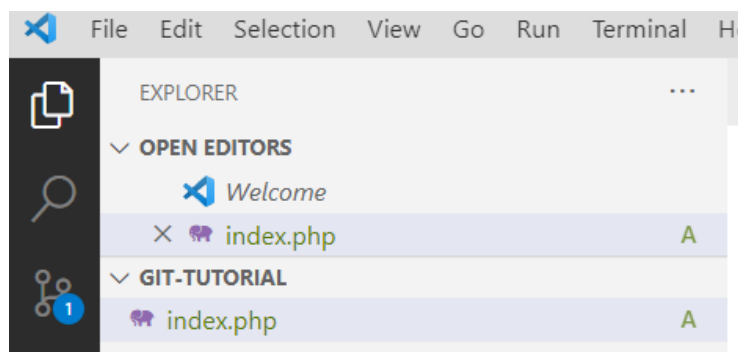
```
nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git add index.php
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git status
On branch master
```

No commits yet

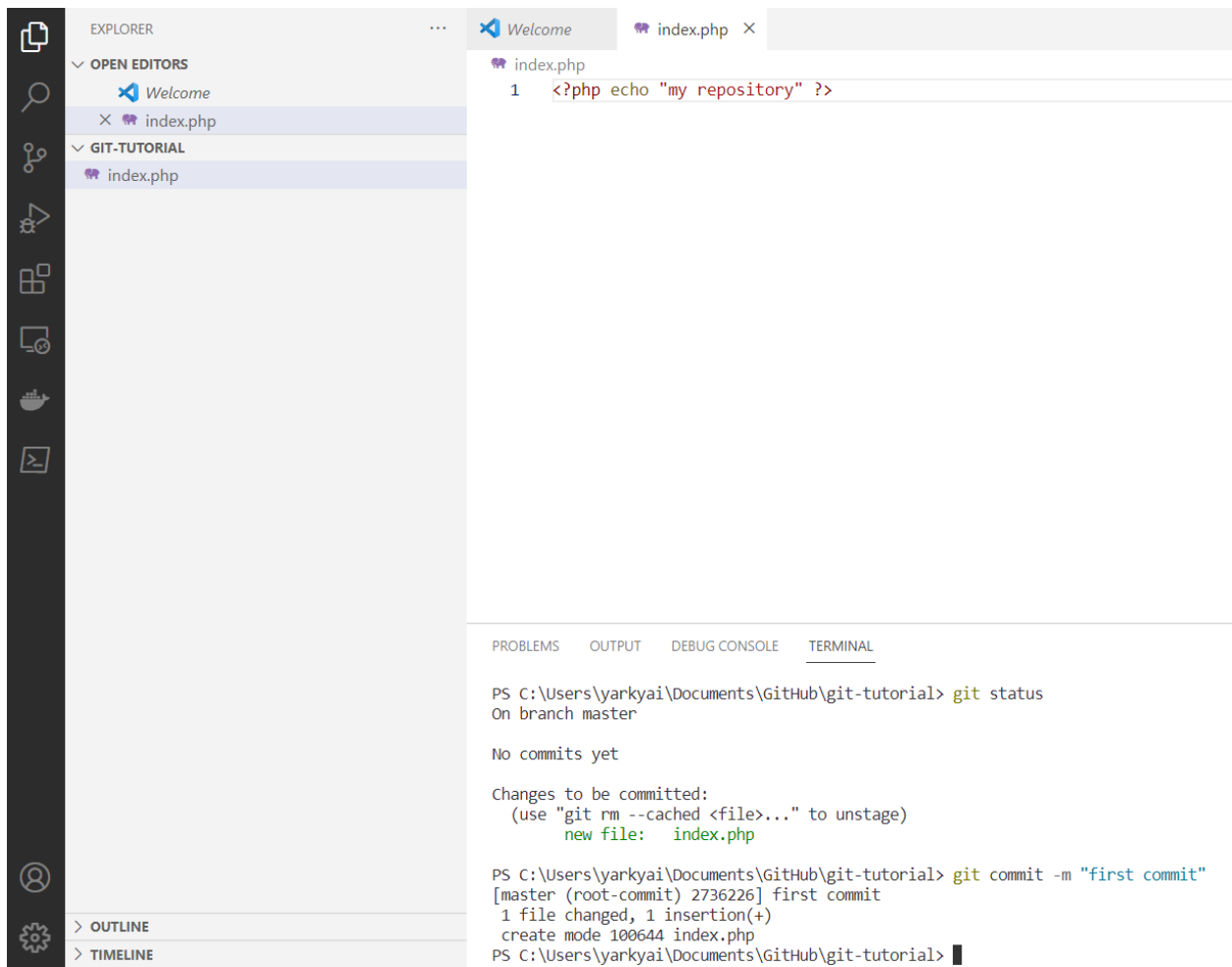
```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.php
```

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

จากรูปจะเห็นว่าเราใช้คำสั่ง git add index.php เพื่อ add ไฟล์ของเราเพื่อเตรียมเข้าสู่ repository จากนั้นลองพิมพ์ git status เพื่อดูสถานะของไฟล์ หรือเราสังเกตที่ VS code ไฟล์ของเราจะมีสถานะเป็น A



จากขั้นตอนนี้ไฟล์ของเราจะยังไม่ถูก save ไปที่ repository นะครับเราต้องพิมพ์คำสั่งอีก 1 คำสั่งคือการ commit นั่นเอง โดยให้เราใช้คำสั่ง git commit -m "frist commit"



จากรูปจะเห็นว่าสถานะของไฟล์นั้นหายไปแล้ว หมายความว่าตัวไฟล์ได้ถูก commit เข้า repository เรียบร้อยแล้ว ทีนี้ถ้าเราอยากรู้กระบวนการทำงานการ commit ย้อนหลังเราก็สามารถใช้คำสั่ง `git log` ได้ ซึ่งโปรแกรมจะแสดงข้อมูลดังนี้

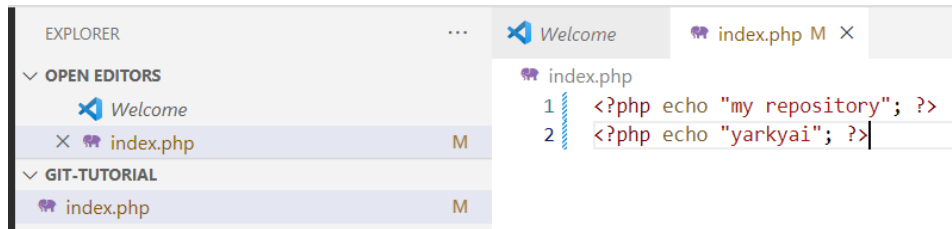
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

  (use "git rm --cached <file>..." to unstage)
        new file:   index.php

PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git commit -m "first commit"
[master (root-commit) 2736226] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 index.php
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git log
commit 273622620d126c3df1c806f3ed99cc4d09fee396 (HEAD -> master)
Author: yarkyai17 <yarkyai17@gmail.com>
Date:   Sun Jul 9 16:54:57 2023 +0700

    first commit
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```


ทดลองเพิ่มคำสั่งเข้าไปใน index.php ของเราก่อนให้ save file แล้วสังเกตผล



พิมพ์ git status โปรแกรมจะบอกว่าการแก้ไขเปลี่ยนแปลงไฟล์

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.php

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

ให้ นศ. ทำการ add และ commit อีกครั้ง จากนั้นพิมพ์คำสั่ง git log (สังเกตเลข commit ดีๆ นะ
เดี่ยวเราจะใช้เลขนี้กัน)

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git add index.php
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git commit -m "second commit"
[master 366186c] second commit
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git log
commit 366186cd3cf02c339405d79b46cab5484fa07ae1 (HEAD -> master)
Author: yarkyai17 <yarkyai17@gmail.com>
Date:   Sun Jul 9 17:06:32 2023 +0700
```

second commit

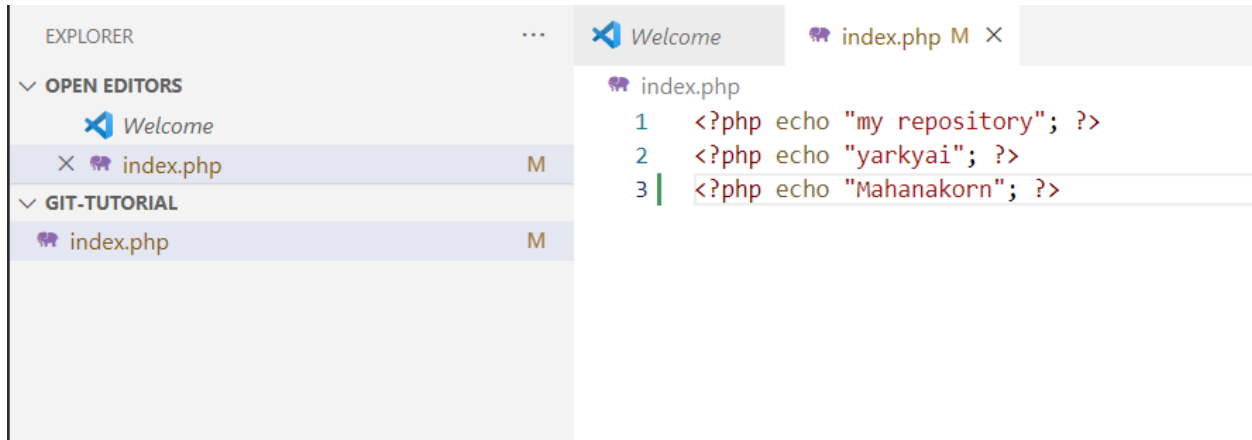
```
commit 273622620d126c3df1c806f3ed99cc4d09fee396
Author: yarkyai17 <yarkyai17@gmail.com>
Date:   Sun Jul 9 16:54:57 2023 +0700
```

first commit

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

ปัญหาหลักๆ ของ Programming คือ เมื่อเราพิมพ์ code ไปเรื่อยๆ เราจะพบว่า เอ้ย เรากลับไปใช้ version เก่าดีกว่า จริงๆ ก็สามารถทำได้ด้วยการ Ctrl Z แต่ถ้าเราแก้ไขไปเยอะเป็นร้อยบรรทัดละ จะต้องกดกี่ครั้ง ที่นี้เราจะเห็นความสำคัญของพวก Version control ละ

ทดลองแก้ไขไฟล์อีกครั้ง และทำการ save อีกครั้ง จะเห็นว่าไฟล์จะขึ้นสถานะ M แล้วเราเอาไฟล์ไปลอง แล้วพบว่าเราย้อนกลับไป version แรกดีกว่า



จากเหตุการณ์ดังกล่าวเราสามารถใช้อคำสั่งได้ดังนี้

- พิมพ์คำสั่ง git log จะเห็นเลข commit ทั้งสองครั้งของเรา

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git log
commit 366186cd3cf02c339405d79b46cab5484fa07ae1 (HEAD -> master)
Author: yarkyai17 <yarkyai17@gmail.com>
Date: Sun Jul 9 17:06:32 2023 +0700
```

second commit

```
commit 273622620d126c3df1c806f3ed99cc4d09fee396
Author: yarkyai17 <yarkyai17@gmail.com>
Date: Sun Jul 9 16:54:57 2023 +0700
```

first commit

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```

- ทำการ copy ตัวเลขมาซัก 6-7 ตัวอักษร แล้วใช้คำสั่ง `git reset --hard` แล้วตามด้วยเลขที่เรา copy มาเมื่อกี้

```

1 <?php echo "my repository"; ?>
2 <?php echo "yarkyai"; ?>

```

```

1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git log
commit 366186cd3cf02c339405d79b46cab5484fa07ae1 (HEAD -> master)
Author: yarkyai17 <yarkyai17@gmail.com>
Date: Sun Jul 9 17:06:32 2023 +0700

second commit

commit 273622620d126c3df1c806f3ed99cc4d09fee396
Author: yarkyai17 <yarkyai17@gmail.com>
Date: Sun Jul 9 16:54:57 2023 +0700

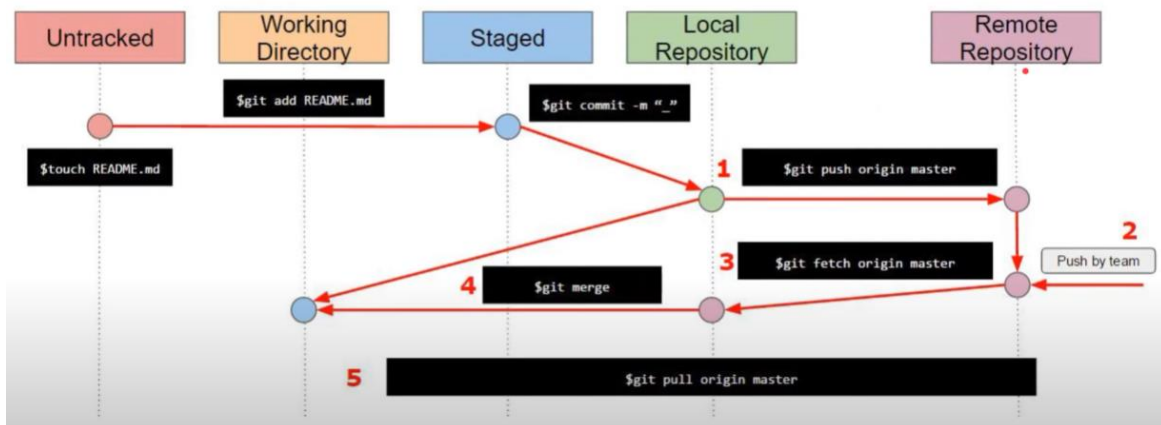
first commit
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git reset --hard 366186cd3cf
HEAD is now at 366186c second commit
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>

```

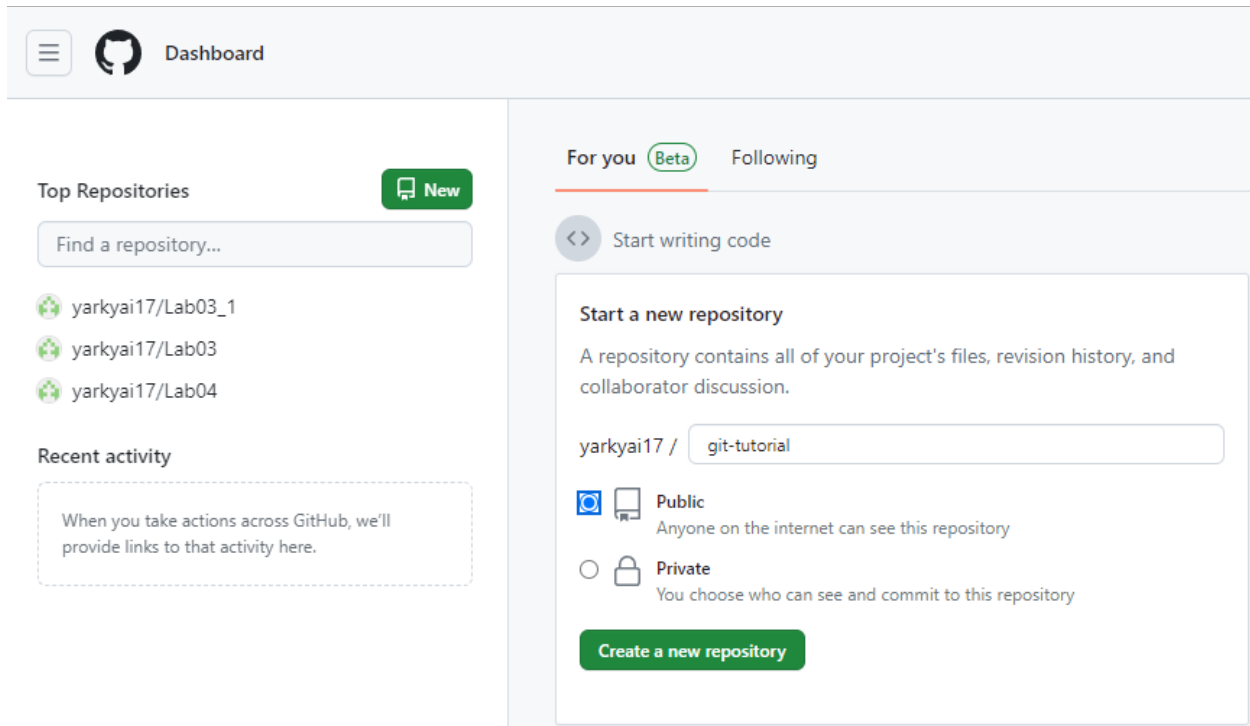
เมื่อใช้คำสั่ง `git reset` โปรแกรมก็จะย้อนกลับไปเวอร์ชันที่เราต้องการได้

การทำงานร่วมกับ Github

จากการทดลองข้างต้นเราจะเป็นการทำอยู่ในฝั่ง local repository เท่านั้น ซึ่งถ้าเราต้องการจะ push ไปที่ฝั่ง remote repository (github) ก็สามารถทำได้ด้วยคำสั่งดังต่อไปนี้



ไปที่เว็บ github สร้าง repository ชื่อว่า git-tutorial (พยายามตั้งให้เหมือนกับเครื่องของเราที่ทำไว้ข้างต้น)



สังเกตสองจุดนี้ไว้ดีๆ เดี่ยวเราจะต้องใช้

กลับมาที่ VS code พิมพ์คำว่า git remote -v เพื่อดูว่าเราได้เชื่อมต่อกับใครไว้หรือไม่ ในที่นี้จะไม่ขึ้นอะไร เพราะเรายังไม่ได้เชื่อมกับ remote ใดๆ เลย

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git remote -v
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```

ต่อมาเราจะเชื่อมต่อกับ remote ละ ให้เราไป copy ข้อมูลที่หน้าเว็บ ที่เป็นคำสั่ง

```
git remote add origin https://github.com/yarkyai17/git-tutorial.git
```

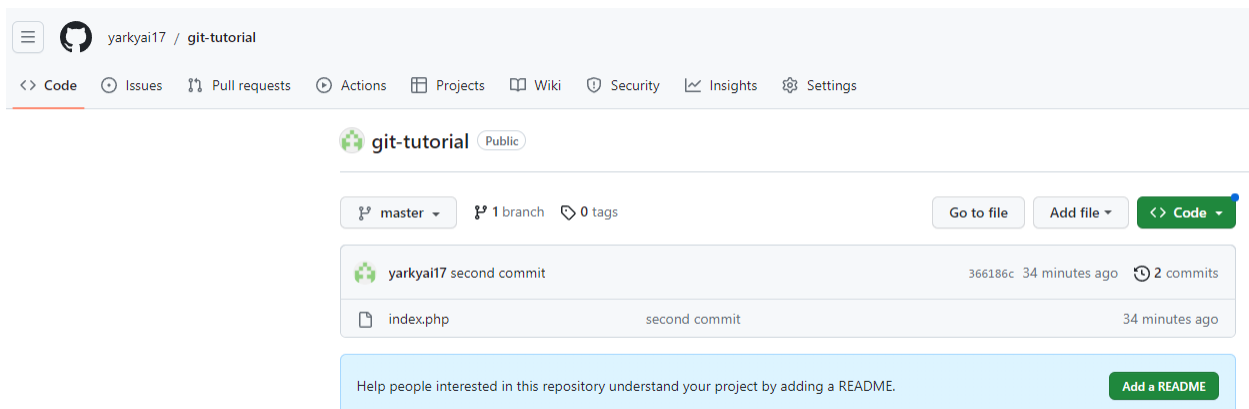
จากนั้นให้เอาไปแปะใน VS code ของเรา แล้วใช้คำสั่ง git remote -v อีกครั้ง จะเห็นว่าเรื่องเราได้ remote ไปที่ปลายทางละ

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git remote add origin https://github.com/yarkyai17/git-tutorial.git
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git remote -v
origin https://github.com/yarkyai17/git-tutorial.git (fetch)
origin https://github.com/yarkyai17/git-tutorial.git (push)
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```

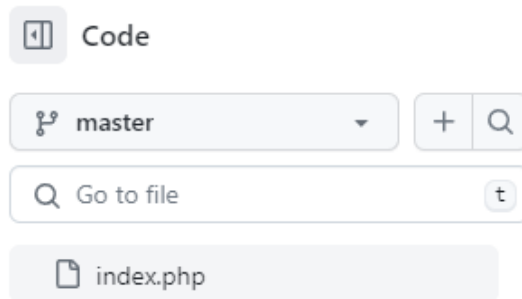
ในตอนนี้เครื่องเราจะต่อกับ remote แล้วแต่ไฟล์ในเครื่องของเราจะยังไม่ไปนะครับ ต้องใช้คำสั่ง git push origin master เพื่อทำการ upload file เราไปที่ github

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git push origin master
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 488 bytes | 488.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/yarkyai17/git-tutorial.git
 * [new branch]      master -> master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```

หลังจากพิมพ์คำสั่ง โปรแกรมจะทำงานให้เราออกซัฟฟัก (มีกรณีที่โปรแกรมจะถาม user pass web github ให้เรายืนยันการ login อีกครั้งก็ให้ใส่ password เว็บ github ไป) กลับไปที่หน้าเว็บ กด refresh จะเห็นว่าไฟล์ index.php ของเรามาละ



กดเข้าไปดู code ได้ดังนี้



git-tutorial / index.php

yarkyai17 second commit

Code Blame 2 lines (2 loc) · 55 Bytes

```
1 <?php echo "my repository"; ?>
2 <?php echo "yarkyai"; ?>
```

คำสั่ง push

ต่อมาเมื่อเราทำการแก้ไขไฟล์ใน VS code แล้วต้องการ push ไปที่ remote ก็สามารถใช้คำสั่งที่ได้เรียนไปเบื้องต้นแล้วดังนี้

```
index.php
1 <?php echo "my repository"; ?>
2 <?php echo "yarkyai"; ?>
3
4 <?php echo "Computer Engineering"; ?>
5 <?php echo "Mahanakorn"; ?>
```

ใช้คำสั่งทั้งหมดอีกครั้งตามรูป

```
index.php
1  <?php echo "my repository"; ?>
2  <?php echo "yarkyai"; ?>
3
4  <?php echo "Computer Engineering"; ?>
5  <?php echo "Mahanakorn"; ?>
```


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL


```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git add index.php
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git commit -m "update to remote"
[master 4b0b8bc] update to remote
1 file changed, 4 insertions(+), 1 deletion(-)
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 320 bytes | 320.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/yarkyai17/git-tutorial.git
366186c..4b0b8bc master -> master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

จากนั้นกลับไปดูไฟล์ที่ตัว github สังเกตว่าจะมีการเปลี่ยนแปลงตามที่เราได้ทำได้



Commit



update to remote

 master

 yarkyai17 committed 1 minute ago

Showing 1 changed file with 4 additions and 1 deletion.

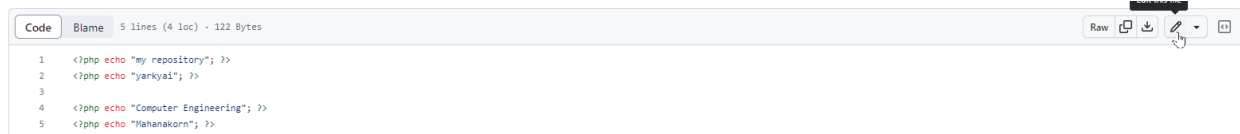
5  index.php 

...	...	@@ -1,2 +1,5 @@
1	1	<?php echo "my repository"; ?>
2	-	<?php echo "yarkyai"; ?>
		
	2	+ <?php echo "yarkyai"; ?>
	3	+
	4	+ <?php echo "Computer Engineering"; ?>
	5	+ <?php echo "Mahanakorn"; ?>
		

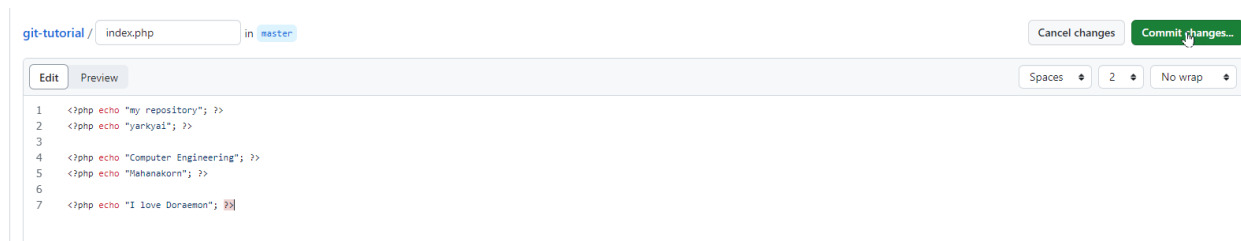
0 comments on commit 4b0b8bc

คำสั่ง git pull

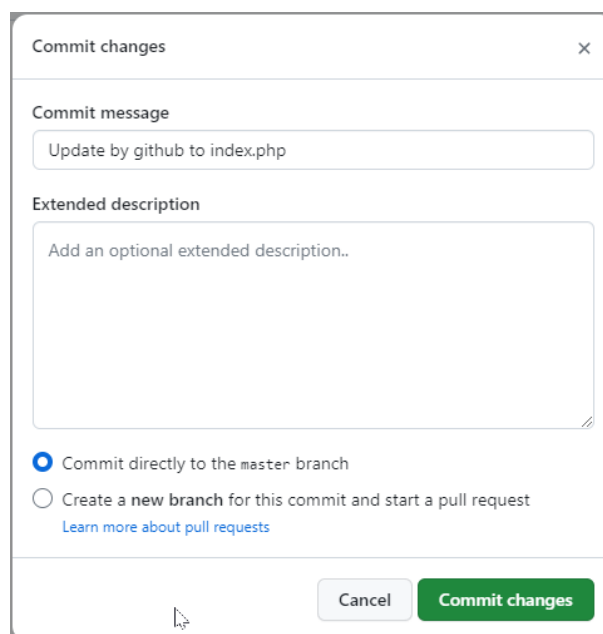
ถ้าไฟล์ของเรามีการทำงานเป็นทีม และเพื่อนร่วมทีมของเรามีการแก้ไขไฟล์ที่เราได้ push ไป ถ้าเราจะเขียนโปรแกรมต่อ ก็ต้องมีการ pull ไฟล์ใหม่มาเพื่อเขียนโปรแกรมต่อ ในที่นี้ผมจะทำแบบง่ายๆ ก่อนนะครับ ให้เราทดลองแก้ไขไฟล์จาก github ได้เลยโดยกดปุ่มแก้ไข



เพิ่มโค้ด บรรทัดที่ 7 เข้าไป แล้วกดปุ่ม Commit changes..

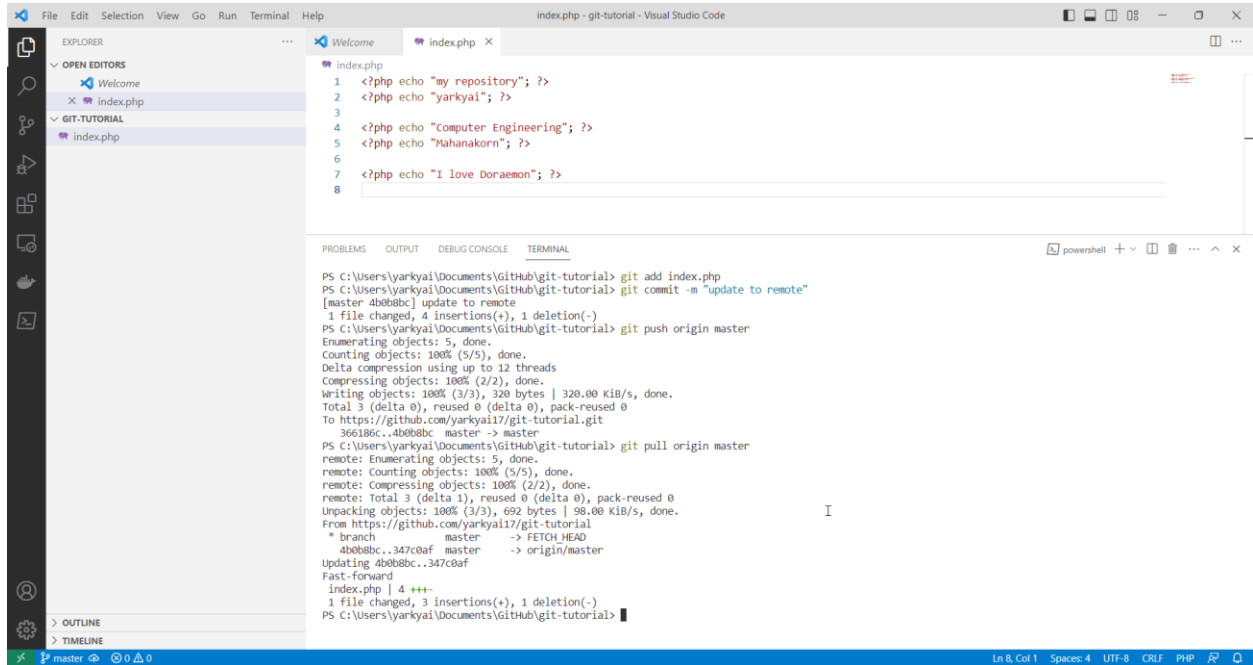


ใน commit message ให้เราพิมพ์ Update by github to index.php แล้วกด Commit changes อีกครั้ง



ในตอนนี้ Code ของเรากับใน remote จะไม่เหมือนกันละ (เราจะไม่สามารถ push ได้ เดี่ยวจะสอนในอาทิตย์หน้า) หากเราจะทำการแก้ไขไฟล์เราจะต้อง pull Code จาก github มาก่อนด้วยคำสั่งดังนี้

`git pull origin master` จะเห็นว่า code ของเราจะเปลี่ยนตาม git hub ละครับ



```
1 <?php echo "my repository"; ?>
2 <?php echo "yarkyai"; ?>
3
4 <?php echo "Computer Engineering"; ?>
5 <?php echo "Mahanakorn"; ?>
6
7 <?php echo "I love Doraemon"; ?>
8

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

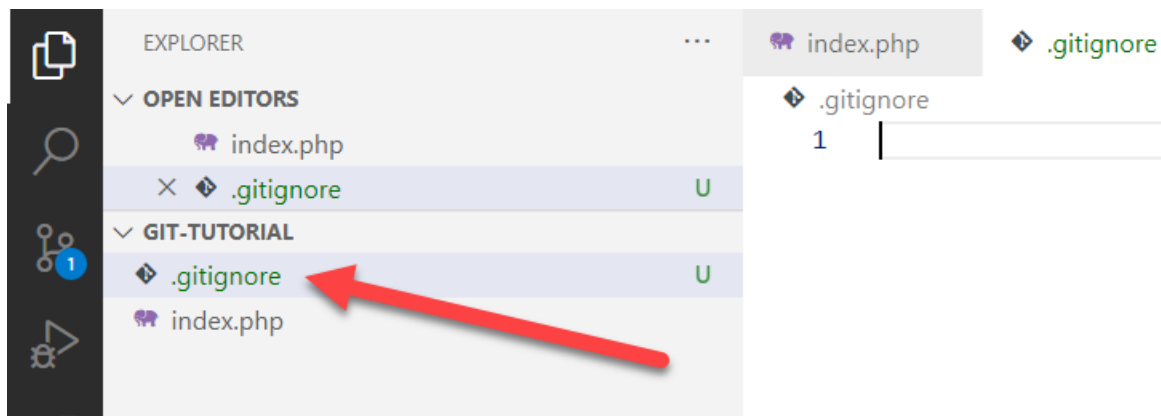
PS C:\Users\yarkyai\Documents\Github\git-tutorial> git add index.php
PS C:\Users\yarkyai\Documents\Github\git-tutorial> git commit -m "update to remote"
[master 4b0b8bc] update to remote
1 file changed, 4 insertions(+), 1 deletion(-)
PS C:\Users\yarkyai\Documents\Github\git-tutorial> git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 320 bytes | 320.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/yarkyai17/git-tutorial.git
366186c..4b0b8bc master -> master
PS C:\Users\yarkyai\Documents\Github\git-tutorial> git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 692 bytes | 98.00 KiB/s, done.
From https://github.com/yarkyai17/git-tutorial
* branch      master       -> FETCH_HEAD
4b0b8bc..347c0af master -> origin/master
Updating 4b0b8bc..347c0af
Fast-forward
 index.php | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)
PS C:\Users\yarkyai\Documents\Github\git-tutorial>
```

Gitignore

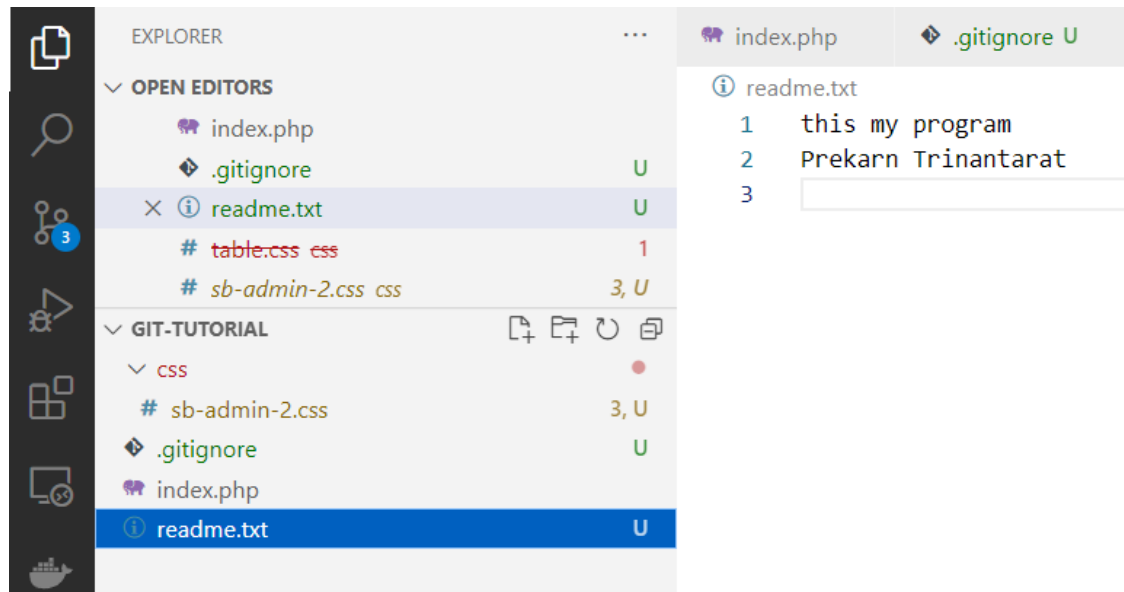
ในบางกรณีที่เราไม่ต้องการให้ git เอาไฟล์จากเครื่องของเรา push ไปยัง git hub เช่น ไฟล์บางไฟล์ที่ไม่มีการแก้ไข ประมาณว่า readme.txt หรือ โฟลเดอร์ที่เป็นประเภทรูปภาพ (ปกติก็ upload รูปภาพที่ใช้กับทีมขึ้น Google drive แล้วให้ทีมมา download ไปแทน) หรือไม่ก็เป็นโฟลเดอร์ที่ใช้เก็บพวกไฟล์ CSS ต่างๆ ที่ไม่ได้มีการแก้ไขใดๆ ดังนั้นในหัวข้อนี้เราจะมาลองใช้คำสั่ง gitignore ที่เป็นการบอกให้เครื่องว่า เวลา push ไม่ต้องเอาไฟล์เหล่านี้ไปนะ

วิธีการใช้ Git Ignore สามารถทำได้ดังนี้

- สร้างไฟล์ .gitignore

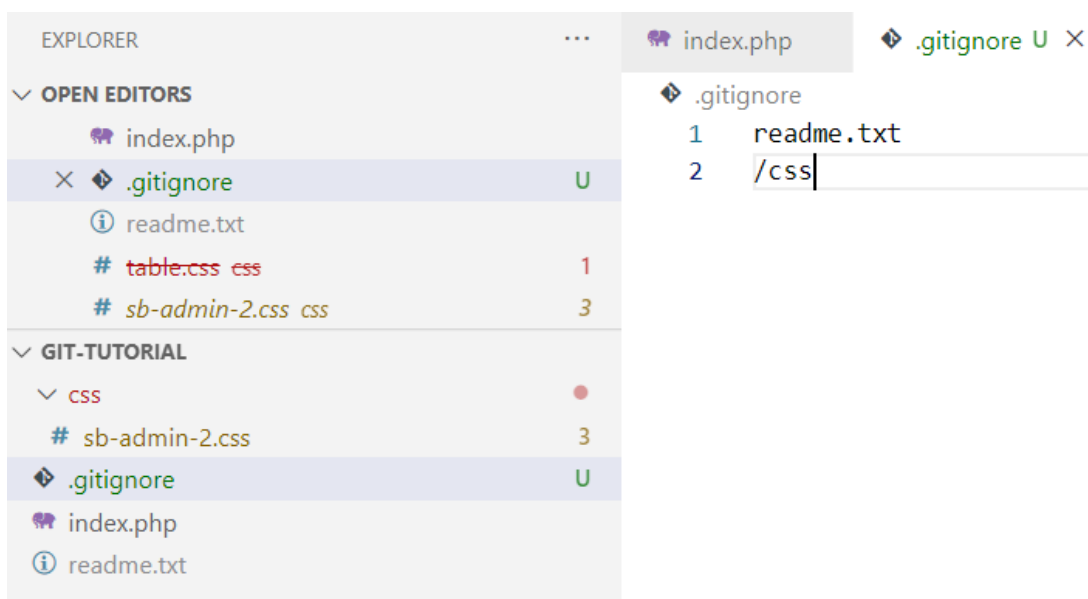


- ทดลองสร้างไฟล์และโฟลเดอร์ใน repository ของเรา



จากรูปด้านบน ให้ทำการสร้างไฟล์ readme.txt เขียนอะไรลงไปก็ได้สักสองบรรทัดแล้ว save จะเห็นว่า มีตัว U ขึ้นหลังไฟล์ จากนั้นก็สร้างโฟลเดอร์ css และให้สร้างไฟล์ในโฟลเดอร์ (ในที่นี้ผมไปก๊อปปี้ไฟล์จากที่อื่นมาใส่ ส่วน นศ. จะเอาไฟล์อะไรมาใส่ก็ได้ เพราะจะอย่างไรไฟล์ก็จะได้ถูกสนใจอยู่แล้ว)

- ในไฟล์ .gitignore ให้พิมพ์ชื่อไฟล์ หรือโฟลเดอร์ที่ต้องการลงไป สังเกตว่าที่ไฟล์ readme.txt ตัว U ด้านหลังจะหายไป



- ทำการ Add และ Commit ไฟล์ .gitignore เข้า Git


```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git add .gitignore
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git commit -m "add .gitignore"
[master 8659aa0] add .gitignore
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

T


- ทดสอบใช้คำสั่ง git push สังเกตว่าไฟล์ที่ push ขึ้นไปมีแต่ไฟล์ .gitignore ส่วนไฟล์ readme.txt และไฟล์เดอร์ css จะไม่ถูก push ไปด้วย


PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL


```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 557 bytes | 557.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/yarkyai17/git-tutorial.git
e663b54..103a007 master -> master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```


 yarkyai17 / git-tutorial

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

 **git-tutorial** Public Pin Unwatch 1

 master


 1 branch

 0 tags


[Go to file](#)

[Add file](#)

[Code](#)


 yarkyai17 lab1607

103a007 1 minute ago 8 commits

 .gitignore

add .gitignore

4 minutes ago

 index.php

lab1607

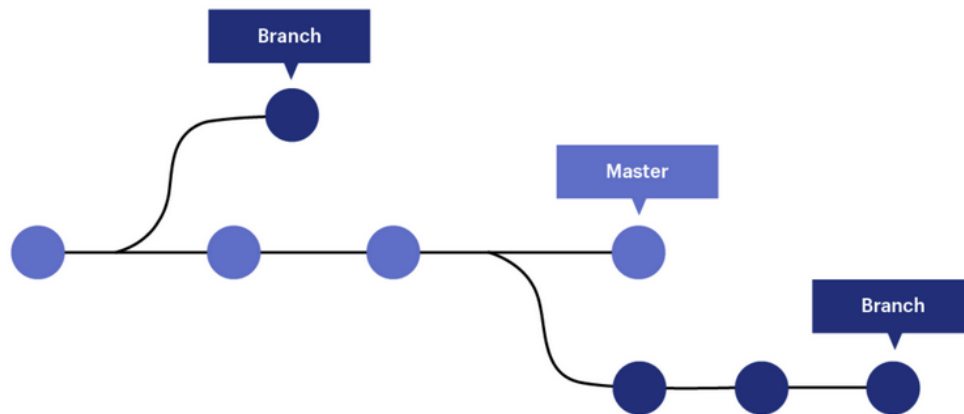
1 minute ago

Help people interested in this repository understand your project by adding a README.

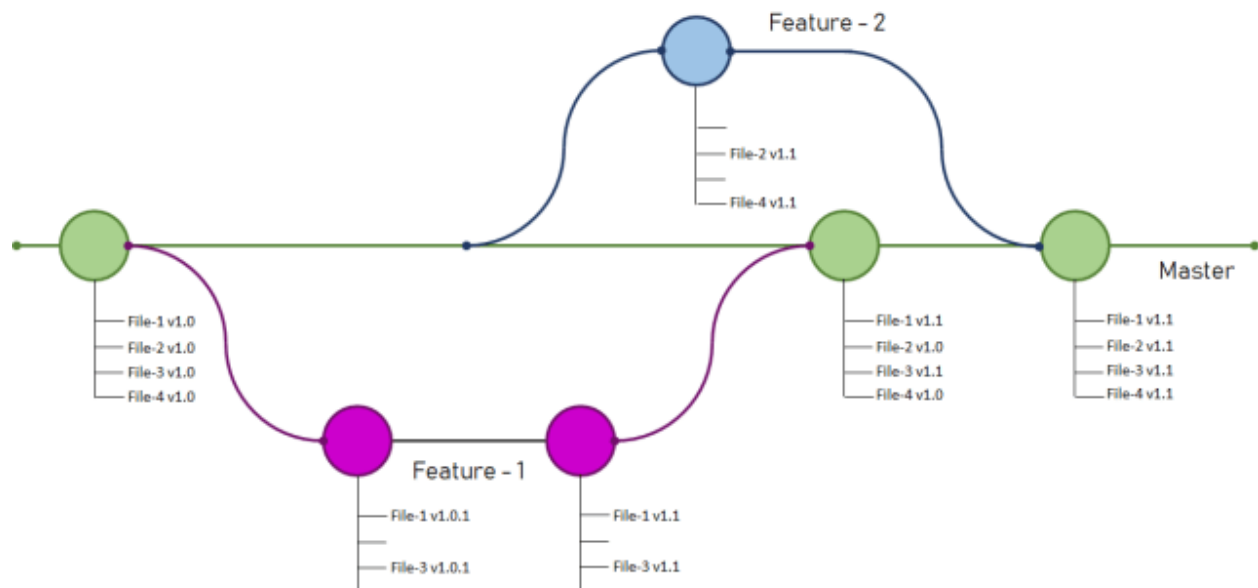
[Add a README](#)

Git Branch/ Merge

ในการใช้งานพวก Version controls ต่างๆนั้นเราสามารถที่จะแตก Branch ออกเป็นเรื่อยๆได้ เช่น เอาไว้เพิ่มความสามารถใหม่ๆ (Feature branch), เอาไว้แก้บัค (HotFix), เอาขึ้น production (Release) ซึ่งใครอยากจะแตก branch ก็แตกได้ทุกที่ทุกเวลา ดังนั้นมันเลยไม่มีกฎตาย และไม่ได้ทำให้อะไรพัง



ข้อดีของการแตก Branch คือ คนในทีมแต่ละคนก็แยกพัฒนาแต่ละ Feature ไปเลย ทดสอบ ทดลองอะไรต่อมิอะไรไป เมื่อมั่นใจว่างานของเราสมบูรณ์แล้วก็ทำการ Merge เข้าไปที่ Master เพื่อใช้งานจริงต่อไป



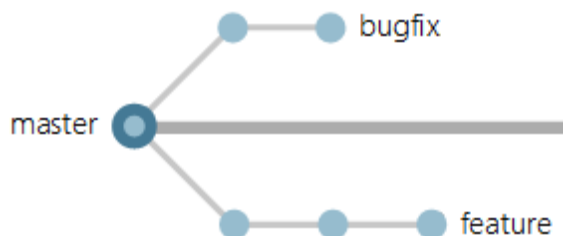
Feature/Topic branches

ถ้าจะเพิ่มความสามารถใหม่ๆ ให้โค้ดเราจะต้อง แยก branch ใหม่จาก master มาทำงานเรื่องนั้นๆ โดยเฉพาะ ซึ่งเราเรียกมันว่า Feature branch หรือบางสำนักก็เรียกว่า Topic branch นั่นเอง

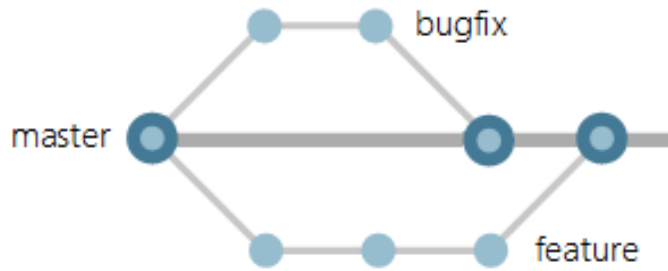
✗ ไม่ควรทำ - หากใครที่ทำงานและ commit ลงใน master branch เลยแบบนี้จะมีปัญหา เพราะไม่มีใครการันตีได้ว่าสิ่งที่ commit เข้ามามันจะไม่เกิด bug และเมื่อมันเข้า master ไปแล้วมันจะเอาออกหรือถอยกลับยากตามรูปด้านล่าง



✓ ควรทำ - ทุกครั้งที่จะมีการเพิ่ม feature ใหม่ๆ หรือแก้บัคอะไรก็ตาม ให้ทำการแตก branch ใหม่ออกมาจาก master ชะ แล้วทำงานที่ branch ใหม่ตัวนั้นแทน เพราะต่อให้เราแก้จนมันพัง หรือไม่อย่างได้โค้ดเหล่านี้แล้ว อย่างมากก็แค่ลบ branch ตัวนี้ออกทุกอย่างก็จบ



จะแก้โค้ดอะไรก็ตามแต่ จงแตก branch ใหม่ แล้วเมื่อไหร่ก็ตามที่เราทดสอบเจ้า branch ใหม่จนมันใจแล้วว่ามันใช้งานได้ ไม่มีบัค เราก็แค่รวม branch เหล่านั้นกลับไปที่ตัว master branch ก็เป็นอันจบ และการรวม branch เข้ามาต้องเป็นแบบ **--no-ff** (no fast forward) ด้วยนะจ๊ะ เพราะถ้าวันนึงเราไปเจอบัคที่มากับ branch ใหม่ตัวนี้ เราก็จะย้อนสถานะกลับมาก่อนที่จะ merge ได้ง่ายๆ



มั่นใจแล้วก็แค่ merge กลับมา

เกร็ดความรู้ ข้อดีที่เราแยก branch ใหม่เมื่อเราจะแก้ไขอะไรก็ตาม (ไม่ว่าจะแก้ไขเล็กน้อยแค่ไหนด้วย) มันจะช่วย
ให้ master branch ของเราไม่มีมีขยะปนเขาไปได้ และเมื่อเรารวมเข้า master เรียบร้อยแล้ว เราอยากจะถอย
กลับก็จะทำได้ง่าย เพราะเราเห็นจุด merge ที่ชัดเจนเลยนั่นเอง ซึ่งการที่จะทำแบบนี้ได้อย่างแท้จริง เราจะต้อง
เพิ่ม **Policy** ว่าให้มันทำการ **LOCK master branch** ด้วย เพียงเท่านั้นก็จะไม่มีใคร commit อะไรต่างๆเข้า
master ได้อีกแล้ว

ทำไมต้อง --no-ff ด้วย ?

หลายคนพอเห็นคำสั่ง --no-ff ก็อาจจะ งงๆ หรือ คาใจว่าทำไมต้องใช้ด้วยหว่า? ดังนั้นเราลองมาดูกันว่า
มันต่างกันยังไง สมมุติว่า branch ของเราตอนนี้เป็นแบบรูปด้านล่าง

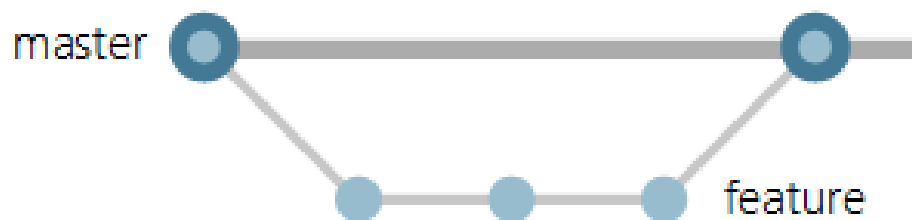


ซึ่งเราต้องการจะเอา feature branch ไปรวมเข้ากับ master branch ด้วยคำสั่ง git merge ซึ่งในบางที่
ถ้าเราใช้ tools ในการ merge หรือจะอะไรก็ตามแต่ ตอน merge เสร็จมันอาจจะเกิดเป็นภาพแบบด้านล่างได้



ปัญหา จริงๆมันก็ไม่ได้ทำให้อะไรเราพังเลยนะ และไม่ได้มีอะไรผิดด้วย แต่เราจะกวาดตามองแล้วไม่รู้เลยว่า commit ไหนบ้างที่เกิดจาก feature นี้ เช่น เรารวม feature Login เข้ากับ master ไปเรียบร้อยแล้ว แล้วมารู้ทีหลังว่า feature Login ที่รวมเข้ามามี bugs ตรึมเลย ทำให้เราต้องย้อนสถานะกลับไปใช้ commit ก่อนที่จะรวม feature Login ซึ่งความสนุกของมันก็คือ commit ไหนล่ะ? เพราะมันอยู่ระนาบเดียวกันหมดเลย ต้องพึ่งความทรงจำของคน commit แล้วละว่าจะจำ message หรือวันเวลาตัวเองได้หรือเปล่านั้นเอง

การ merge โดยใช้คำสั่ง `--no-ff` ควบคุมไปด้วย เป็นการการันตีว่า branch ที่ merge เข้าไปมันจะต้องถูกแยกออกไว้แบบเดิมนั่นเอง และมันจะต้องมี merge commit ด้วยนั่นเอง เลยจะทำให้เราได้ตามภาพด้านล่างเสมอ



Master branch พร้อมใช้งาน

เวลาที่เราจะเพิ่ม features ต่างๆเข้าไป เราจะต้องแตก branch ออกไป ดังนั้นของทุกอย่างที่อยู่ใน master branch มันจะต้องพร้อมใช้งานเสมอ build ได้ไม่มี error และ run test ผ่านทั้งหมด เพราะไม่อย่างนั้น ตอนที่ทีมแตก branch ออกไปเขียน features ต่างๆ ก็จะเจอ error ต่างๆ จนทำให้เสียเวลามาแก้ปัญหาพวกนั้น ก่อนที่จะเริ่มทำงานกันยังงี้ละ

การตั้งชื่อ branch

เวลาที่เราจะแตก branch ออกมาทำอะไรก็ตาม ในทีมควรที่จะมีการตกลงมาตรฐานในการตั้ง branch เหล่านั้นด้วย เพราะพอเวลาผ่านไป เราก็อาจจะจำไม่ได้แล้วว่าเราแตก branch นั้นมาทำไม ? ดังนั้นชื่อของมันจะเป็นสิ่งแรกๆที่จะทำให้เราระลึกชาติได้นั่นเอง เช่น เราเจอชื่อ branch Login1 กับ Login 2 เราจะรู้ใหม่ว่ามันต่างกันตรงไหน? ซึ่งทาง Microsoft เขามีหลักในการตั้งชื่อ branch ตามด้านล่างนี้

แบ่งตามคนที่ดูแลงานนั้นๆ

- users/username/description
- users/username/yarkyai

แบ่งตามการแก้ไขผิดพลาด

- bugfix/description
- hotfix/description

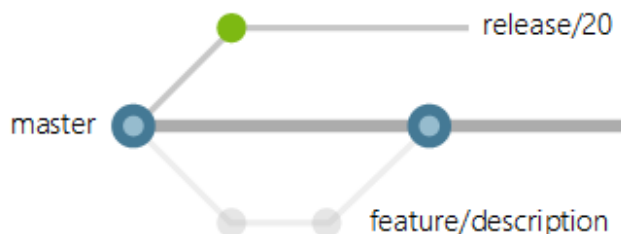
แบ่งตามความสามารถใหม่ๆ

- features/feature-name
- features/feature-area/feature-name
-

มันอาจจะดูยาวๆหน่อย แต่ถ้าคิดดีๆมันก็มีประโยชน์ในรูปแบบของมัน ซึ่งข้อดีในการตั้งชื่อแบบนี้ใน Azure DevOps มันจะช่วยให้เราจัดการเรื่อง [Branch folders](#) ได้ด้วย

Release branches

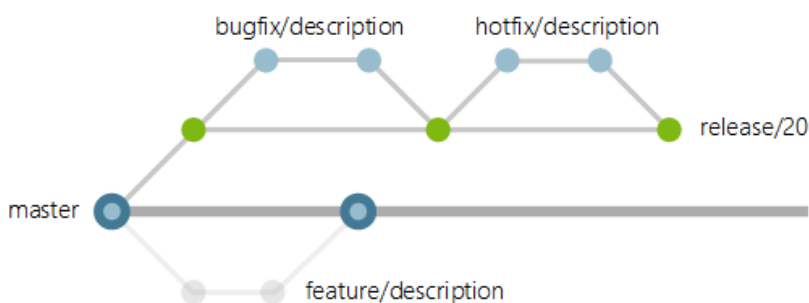
เมื่อเรามั่นใจว่างานที่อยู่ใน master สามารถนำขึ้น production ได้แล้ว เราก็จะเริ่มแตก branch ใหม่ ออกไป ซึ่งเราเรียกมันว่า **release branch** นั่นเอง ซึ่งเราอาจจะตั้งชื่อให้รู้ว่าเป็น release version อะไรลงไป ด้วยก็ได้ ตามรูปด้านล่าง (สีเขียว) หรือบางทีก็เป็นการ release **ตอนจบ Sprint/Iteration**



จะขึ้น production ให้แตก release branch แยกต่างหาก

ซึ่งเจ้าตัว release branch นี้จะต่างจาก feature branch ตรงที่มันจะไม่เคยถูก merge กลับมาที่ master branch เลยยังงั้นละ!! สาเหตุเพราะว่า งานที่ release ไปแล้ว เราจะไม่ไปเพิ่มความสามารถอะไรใหม่ๆ ใน นั้น (พูดง่ายคือไม่แตก feature branch จาก release branch) ดังนั้นเราก็ไม่มีเหตุผลที่จะรวมมันกลับมาที่ master

แต่ถ้าเราพบว่าใน release branch มี bug เกิดขึ้น เราก็สามารถแตก branch ใหม่ออกจาก release เพื่อทำการแก้ bug เหล่านั้นก็รับได้นะ ตามรูปเบย



แก้ bug ที่อยู่ใน release branch

คำสั่ง Branch

- `git branch` คำสั่งแสดง branch ของเราทั้งหมด
- `git branch develop` สร้าง branch ชื่อว่า develop ขึ้นมา
- `git branch -delete develop` ลบ branch ชื่อว่า develop

Git Checkout : สลับการทำงานระหว่าง Branch หรือ Commit ID

```
#ย้ายการทำงานไปที่ Branch หรือ commit_id ที่ระบุ
git checkout <branch name, commit id>

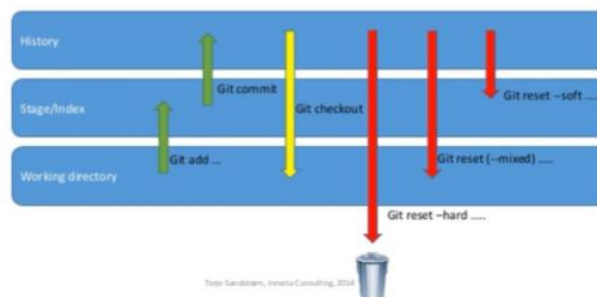
#สร้าง branch ชื่อ test และทำการสลับการทำงานมาที่ Branch นี้
git checkout -b test

#เลือกแค่บางไฟล์จาก Branch อื่น เข้ามา Merge กับ Working Directory ที่กำลังทำงาน
git checkout <branch name> <file name>
```

Git Reset : ย้อนกลับไปก่อนหน้าด้วย Branch หรือ Commit ID

```
git reset --hard d31f399      #ย้อนกลับไปที่ Commit id d31f399 | HARD
git reset --soft d31f399     #ย้อนกลับไปที่ Commit id d31f399 | SOFT
git reset --mixed develop    #ย้อนกลับไปที่ Branch develop | MIXED
```

Git tree movements visualized



มาลองสร้าง branch develop กัน

ใช้คำสั่ง git branch เพื่อดูว่าเรามี branch อะไรบ้าง สร้าง branch develop ด้วยคำสั่ง git branch develop จากนั้นใช้คำสั่ง git branch อีกครั้ง

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch
* master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch develop
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch
develop
* master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

สร้าง branch feature ขึ้นมาอีกอันครับ

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch feature
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch
develop
feature
* master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

ทดลองลบ branch feature กัน ด้วยคำสั่ง git -delete feature

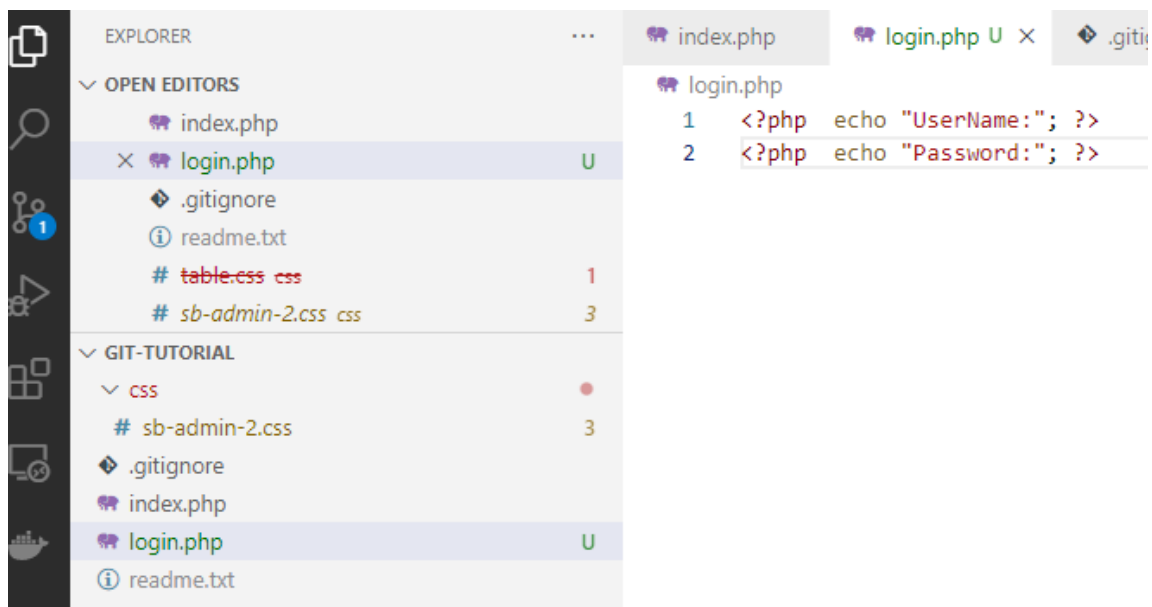
```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch --delete feature
Deleted branch feature (was 103a007).
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch
develop
* master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

หลังจากเราลบได้แล้วจะเห็นว่าเหลือแค่ branch master กับ develop เท่านั้น

ต่อมาเราจะเปลี่ยนมาทำงานที่ branch develop กัน ด้วยคำสั่ง git checkout develop จากนั้นของใช้คำสั่งเพื่อดู branch จะเห็นว่าตัว * จะมาอยู่ที่ develop ละ

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git checkout develop
Switched to branch 'develop'
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch
* develop
  master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

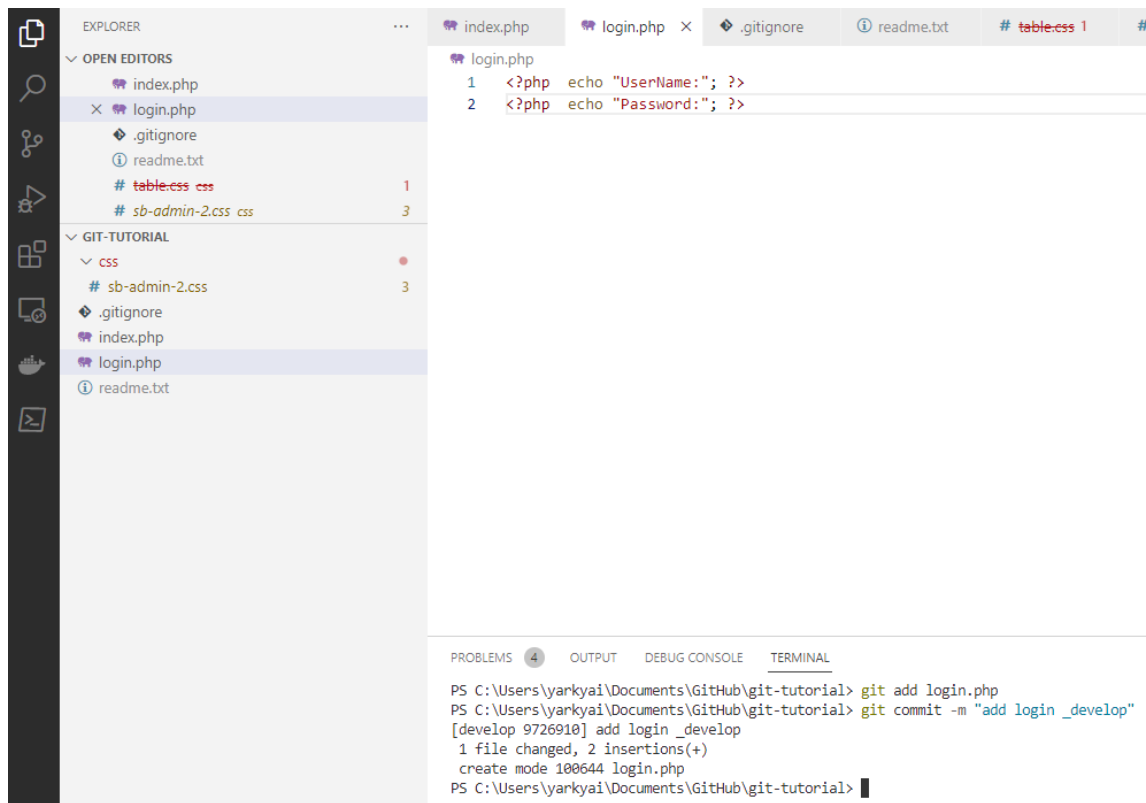
ที่ branch develop ให้สร้างไฟล์ขึ้นมาใหม่ชื่อว่า login.php แล้วเขียนโปรแกรมดังรูป



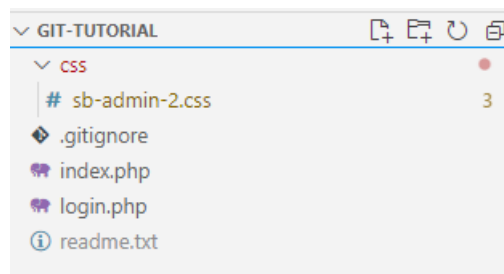
ทำการ add และ commit

```
git add login.php
```

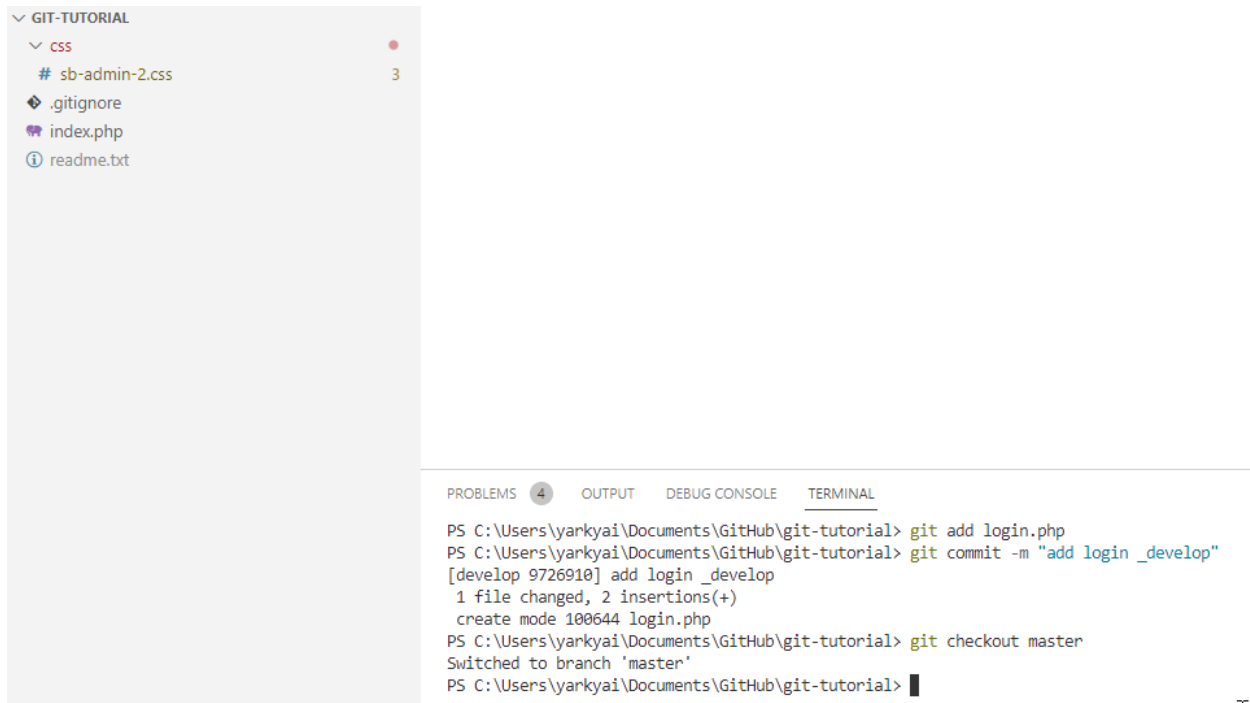
```
git commit -m "add login _develop"
```



สังเกตตรงนี้ดีๆ นะครับเราจะเห็นว่าไฟล์ใน branch ของเรามีไฟล์ index.php และ login.php



ลองสลับกลับไป branch master แล้วดูไฟล์อีกครั้งหนึ่งจะพบว่ามีแค่ไฟล์ index.php เท่านั้น



เรามาทดลอง push ขึ้น git hub กันครับ

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git checkout develop
Switched to branch 'develop'
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch
* develop
  master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git push origin develop
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 352 bytes | 352.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/yarkyai17/git-tutorial/pull/new/develop
remote:
To https://github.com/yarkyai17/git-tutorial.git
 * [new branch]      develop -> develop
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```

ลองดูใน git hub จะมี 2 branch ละ

git-tutorial Public

develop had recent pushes 1 minute ago [Compare & pull request](#)

master 2 branches 0 tags [Go to file](#) [Add file](#) [Code](#)

Your master branch isn't protected
Protect this branch from force pushing or deletion, or require status checks before merging. [Learn more](#) [Protect this branch](#)

yarkyai17 lab1607 103a007 2 hours ago 8 commits

.gitignore	add .gitignore	2 hours ago
index.php	lab1607	2 hours ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

ลองเปลี่ยน branch ดูสิ

git-tutorial Public

develop had recent pushes 3 minutes ago [Compare & pull request](#)

develop 2 branches 0 tags [Go to file](#) [Add file](#) [Code](#)

This branch is 1 commit ahead of master. [Contribute](#)

yarkyai17 add login _develop 9726910 26 minutes ago 9 commits

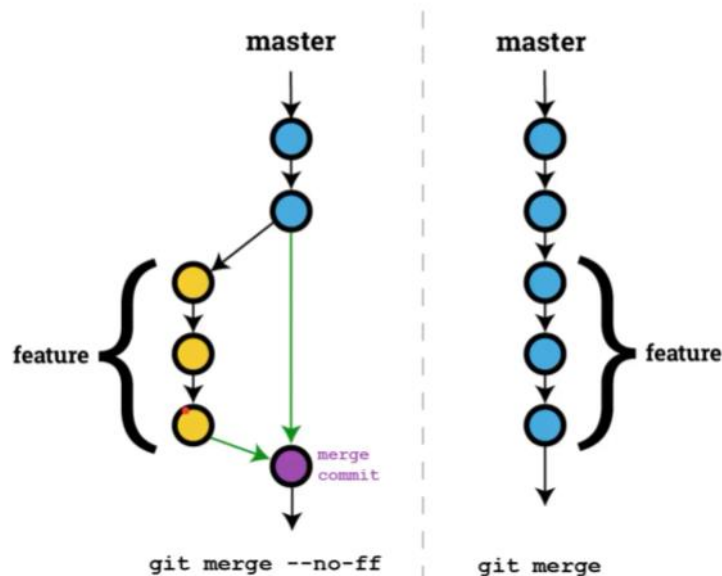
.gitignore	add .gitignore	2 hours ago
index.php	lab1607	2 hours ago
login.php	add login _develop	26 minutes ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

Git Merge

ถึงแม้ว่าเราจะแยก Branch ในการทำงาน แต่สุดท้ายแล้วเราก็ต้องทำการ merge เข้าที่ master อยู่ดี (เนื่องจาก Project ของเราทั้งหมดที่จะ Deploy ขึ้น web เราก็จะเอามาจาก master นี่แหละ) ดังนั้นเราจึงต้องทำการรวมไฟล์จาก branch ต่างๆ มาไว้ที่ master

Git Merge : รวม Branch หรือ Commit ทั้งสองเข้าด้วยกัน



คำสั่งในการ merge เราสามารถใช้ได้ดังนี้

```
#รวม branch master กับ branch feature แบบ no fast forward
$git merge --no-ff feature

#รวม branch master กับ branch feature แบบ fast forward
$git merge feature
```

ถ้าเราต้องการดูความแตกต่างของทั้ง 2 branch เราสามารถใช้คำสั่งได้ดังนี้

Git Diff : เปรียบเทียบ Working Directory กับ Branch หรือ Commit ID ที่ระบุ

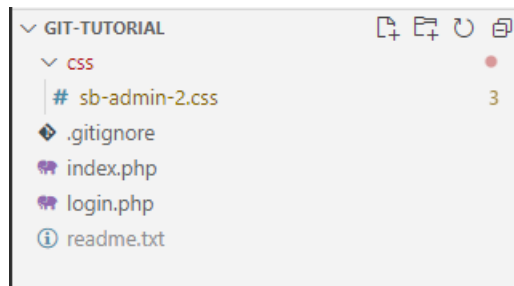
```
git diff 82de188
git diff develop
```

```
To https://github.com/yarkyai17/git-tutorial.git
f367c69..9c88de9 master -> master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git diff develop
diff --git a/index.php b/index.php
index c7ef0fa..1a15a18 100644
--- a/index.php
+++ b/index.php
@@ -9,3 +9,4 @@
 <?php echo "I love Doraemon"; ?>
 <?php echo "I love Doraemon"; ?>
 <?php echo "I love Doraemon"; ?>
+<?php echo "I love Doraemon"; ?>
\ No newline at end of file
diff --git a/login.php b/login.php
deleted file mode 100644
index db0335b..0000000
--- a/login.php
+++ /dev/null
@@ -1,2 +0,0 @@
-<?php echo "UserName:"; ?>
-<?php echo "Password:"; ?>
\ No newline at end of file
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```

ทำทดลอง merge กัน โดยให้เราอยู่ที่ branch master นะ ให้ทุกคนลองเช็คว่าก่อนด้วยคำสั่ง git branch

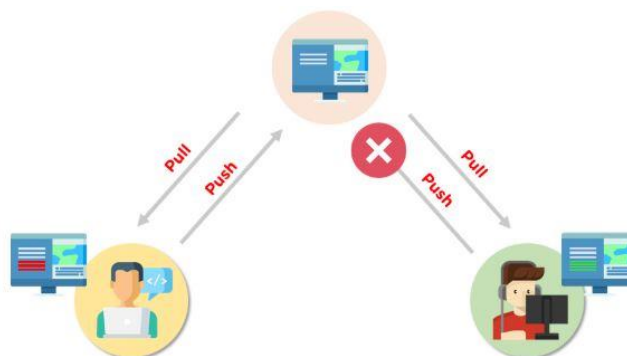
```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git branch
develop
* master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git merge --no-ff develop
Merge made by the 'ort' strategy.
 login.php | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 login.php
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> █
```

จากนั้นก็ใช้คำสั่ง git merge --no-ff develop สังเกตว่าไฟล์ login.php มาอยู่ที่ master ละ



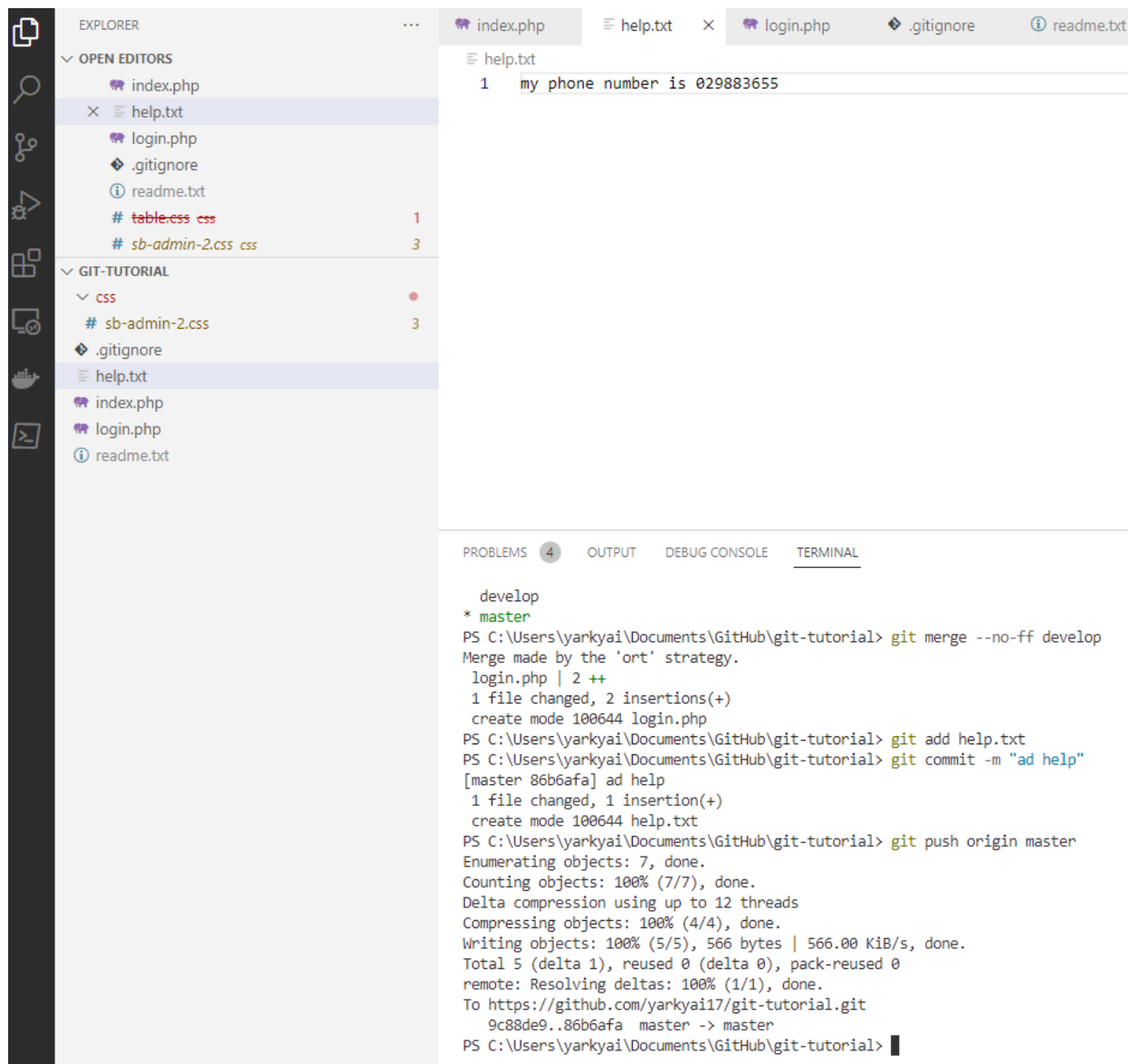
ปัญหา Conflict

ในการทำงานเป็นทีมและมีการทำงานที่ไฟล์ๆ เดียวกัน เช่น นาย สว กับนาย พิธา ทำงานที่ไฟล์ index.php เหมือนกัน ต่อมานาย สว ได้ทำการ push index.php เข้าที่ github ก่อนโดยที่ไม่ได้บอกกับนาย พิธา ไว้ก่อน และเมื่อนาย พิธา แก้ไขไฟล์ index.php ในเครื่องตัวเองเรียบร้อยแล้ว จะทำการ push ขึ้น github ในกรณีนี้ นาย พิธา จะไม่สามารถเป็นนายกได้ เฮ้ยๆ นาย พิธา จะไม่สามารถ push index.php ขึ้นไปได้ ปัญหาแบบนี้เราเรียกมันว่า Conflict



การแก้ไข้ปัญหา สามารถทำได้โดยให้นาย พิธา pull ไฟล์เวอร์ชันล่าสุดมาก่อนแล้วรวมไฟล์ของตัวเองกับไฟล์ของนาย สว เข้าด้วยกัน (ตรงนี้มีหลายแบบนะ เตียวสอนให้) จากนั้นค่อย push ขึ้นไปที่ github อีกครั้ง

มาทดลองกัน ให้ นศ. สร้างไฟล์ใหม่ขึ้นมา 1 ไฟล์ โดยให้ชื่อว่า help.txt และเขียนข้อความดังรูป



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows the file structure of a project named 'GIT-TUTORIAL'. The 'OPEN EDITORS' list includes 'index.php', 'help.txt', 'login.php', '.gitignore', and 'readme.txt'. The 'GIT-TUTORIAL' folder is expanded, showing subfolders 'css' and 'sb-admin-2.css', and files '.gitignore', 'help.txt', 'index.php', 'login.php', and 'readme.txt'. The 'help.txt' file is selected. The main editor area shows the content of 'help.txt', which is '1 my phone number is 029883655'. The bottom panel shows the 'TERMINAL' tab with the following output:

```
develop
* master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git merge --no-ff develop
Merge made by the 'ort' strategy.
 login.php | 2 ++
1 file changed, 2 insertions(+)
 create mode 100644 login.php
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git add help.txt
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git commit -m "ad help"
[master 86b6afa] ad help
1 file changed, 1 insertion(+)
 create mode 100644 help.txt
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 566 bytes | 566.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/yarkyai17/git-tutorial.git
9c88de9..86b6afa master -> master
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```

ทำการ add / commit / push ไฟล์ให้เรียบร้อย

ที่ github จะมีไฟล์ help.txt เรียบร้อยละ

The screenshot shows the GitHub interface for the repository 'yarkyai17 / git-tutorial'. The repository is public and has 2 branches and 0 tags. A notification states 'Your master branch isn't protected'. Below this, a commit by 'yarkyai17' is shown with the message 'ad help'. The commit details table lists the following changes:

File	Commit Message	Time Ago
.gitignore	add .gitignore	4 hours ago
help.txt	ad help	1 minute ago
index.php	230716_1	1 hour ago
login.php	add login _develop	1 hour ago

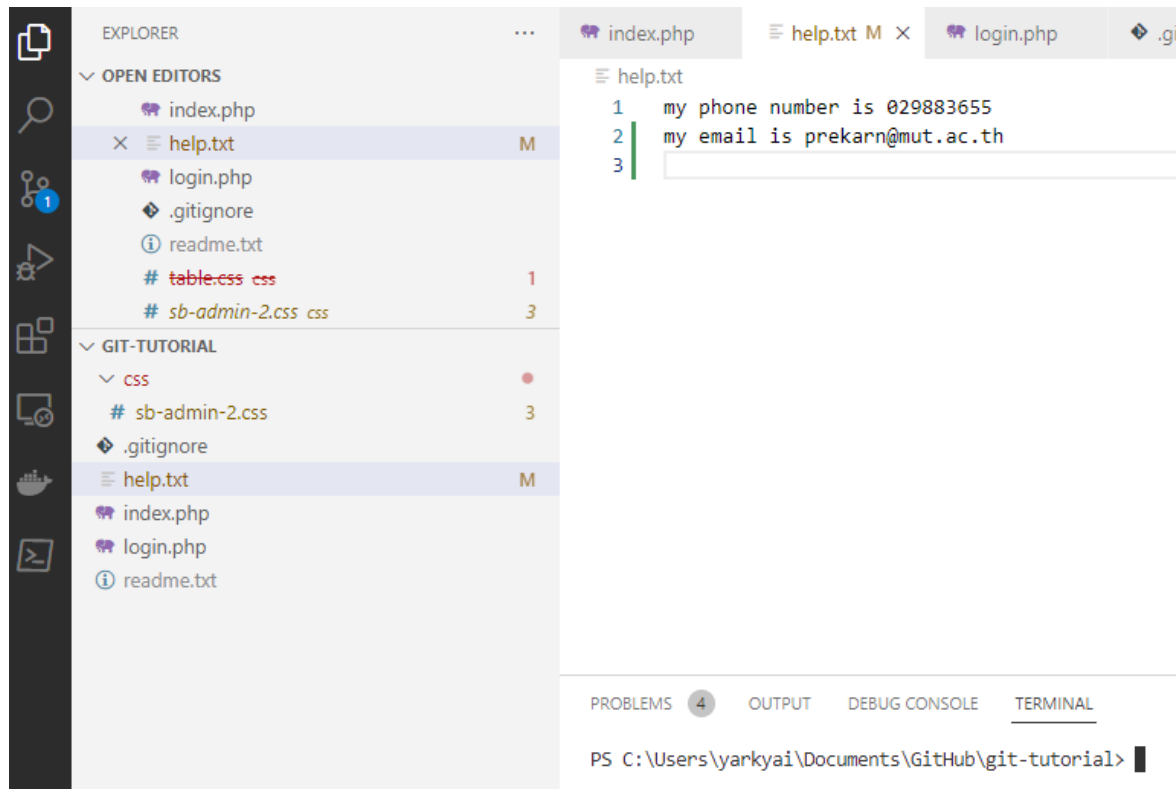
ทำการแก้ไขไฟล์ help.txt ด้วย github และเพิ่มบรรทัดที่สองเข้าไปจากนั้นกด Commit changes ได้เลย ทีนี้ไฟล์ในเครื่องเรา กับไฟล์ที่ github ก็จะไม่เหมือนกันละ

The screenshot shows the GitHub file editor for 'git-tutorial / help.txt' in the 'master' branch. The editor has tabs for 'Edit' and 'Preview'. The 'Edit' tab is active, showing the following content:

```
1 my phone number is 029883655
2 my emial is yarkyai17@gmail.com
```

Buttons for 'Cancel changes' and 'Commit changes...' are visible at the top right. The editor also shows settings for 'Spaces' (2) and 'Soft wrap'.

กลับมาที่เครื่องเราให้เราเพิ่มบรรทัดที่สองในเครื่องเรา เป็น my email is prekarn@mut.ac.th



ทำการ add /commit / push ดูอีกครั้งแล้วสังเกตผลการทำงานว่าเป็นอย่างไร

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git add help.txt
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git commit -m "update help"
[master 6880fa9] update help
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git push origin master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

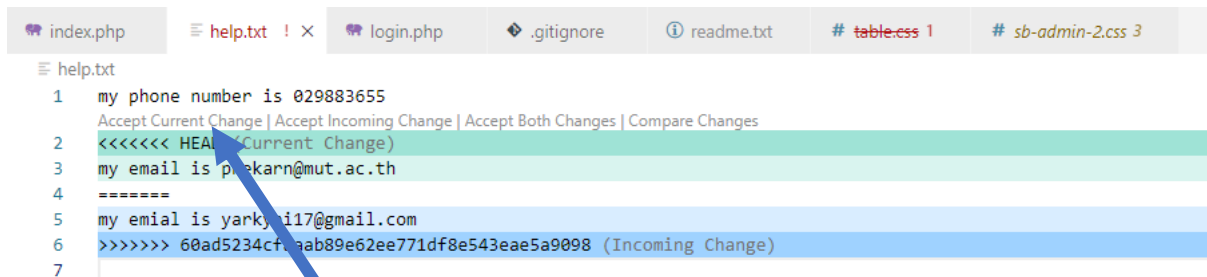
Please make sure you have the correct access rights
and the repository exists.
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git push origin master
To https://github.com/yarkyai17/git-tutorial.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/yarkyai17/git-tutorial.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```

เราจะเห็นว่าไม่สามารถ push ไฟล์ขึ้นไปได้เนื่องจาก ไฟล์ที่เราแก้ไข กับไฟล์ปลายทางมันไม่เหมือนกัน

วิธีการแก้ไขปัญหานี้ทำได้โดยให้เรา pull ไฟล์จาก github มาที่เครื่องเราอีกครั้งหนึ่งก่อน

```
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial> git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 694 bytes | 57.00 KiB/s, done.
From https://github.com/yarkyai17/git-tutorial
* branch      master      -> FETCH_HEAD
   86b6afa..60ad523  master  -> origin/master
Auto-merging help.txt
CONFLICT (content): Merge conflict in help.txt
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\yarkyai\Documents\GitHub\git-tutorial>
```

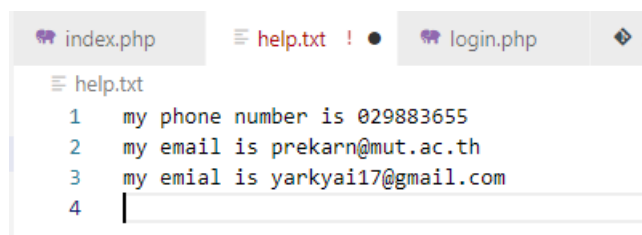
หลังจาก pull มาแล้วจะเห็นว่าตัว VS Code จะแสดงให้เห็นถึงข้อแตกต่างระหว่างไฟล์ทั้งสอง



ในตัว VS Code จะมีข้อความแสดงว่าให้เราเลือกแบบไหนอยู่ด้านบน

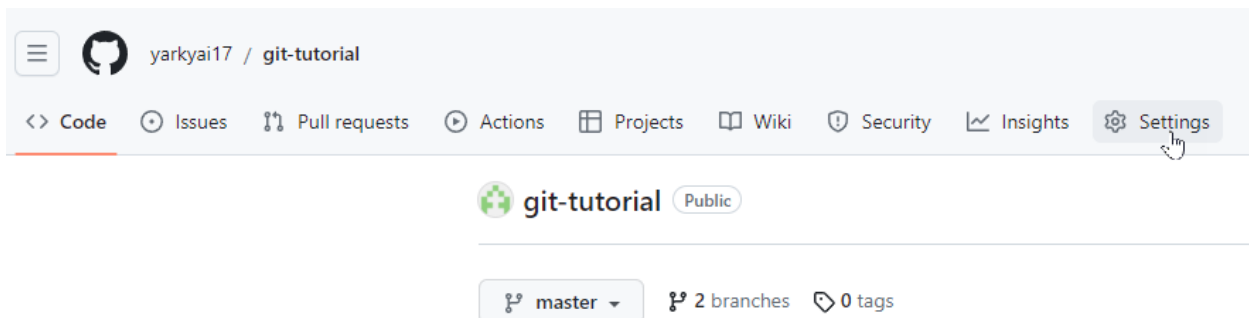
- Accept Current Change ยืนยันว่าของเราถูกและทิ้งของเพื่อนไปซะ
- Accept Incoming Change เออไม่แน่ใจของเรา ของเพื่อนน่าจะถูกแหละ เอาของเพื่อนละกัน
- Accept Both Change เอาทั้งคู่มารวมกันเลย
- Compare Change ดูความแตกต่างของทั้งสองไฟล์

ในที่นี้ผมเลือก Accept Both Change นะครับและผลที่ได้คือ



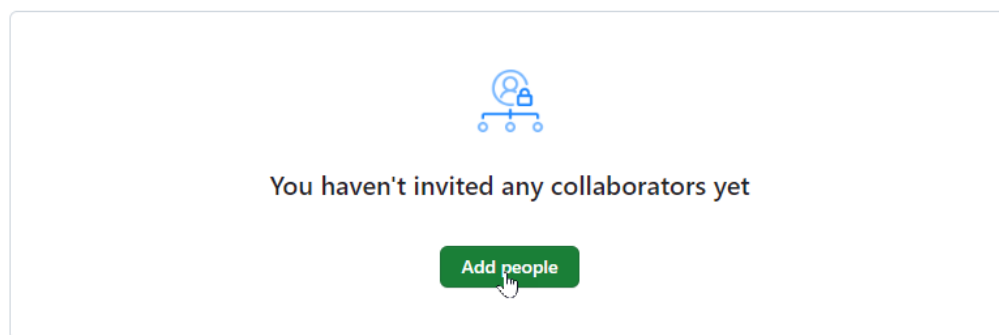
การเพิ่มเพื่อนเข้า Repository ของเรา

- ไปที่ setting

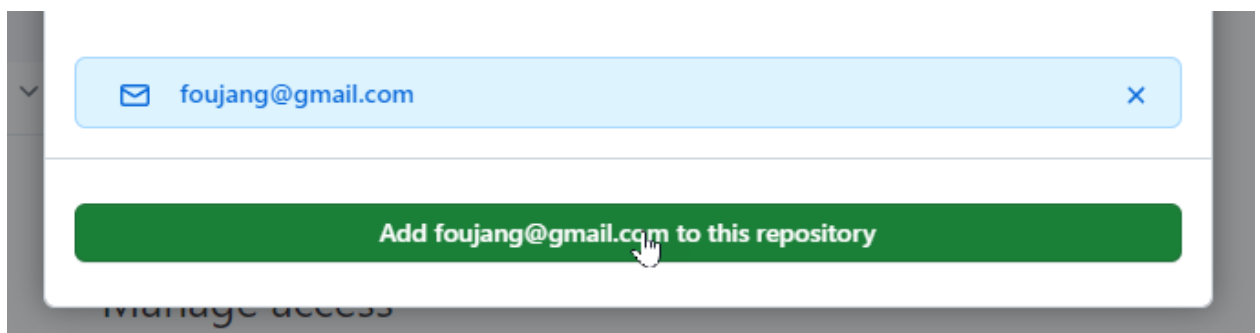


- ไปที่ Collaboration กดปุ่ม Add people

Manage access



- ใส่ email ของเพื่อนเรา



General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules (Beta)

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS


1 has access to this repository. 1 collaborator.

Manage access

[Add people](#)

☐ Select all Type

Find a collaborator...

<input type="checkbox"/>	 foujang Collaborator	Remove
--------------------------	------------------------------------------------------------------------------------------------------------------	------------------------

< Previous Next >



ให้เพื่อนเช็คเมล ในตัวอย่างผมเชิญ อ.โบ มา แล้วพอ อ.โบกดยอมรับ ก็จะสามารถเห็น Repository ของผมได้

Mail 4G 12:35 63%

github.com

yarkyai17 / Lab03

Code Issues Pull requests

 + 

yarkyai17 invited you to collaborate

[Accept invitation](#) [Decline](#)

Owners of Lab03 will be able to see:

- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?
[Block yarkyai17](#)




TRUE-H 4G 12:36 62%

github.com

yarkyai17 / Lab03

Code Issues Pull requests

You now have push access to the yarkyai17/Lab03 repository.

test git lab


0 stars 0 forks 1 watching

Activity

Public repository

main

Branches Tags

 **yarkyai17** last week