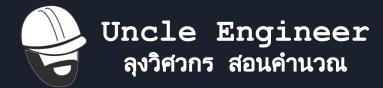


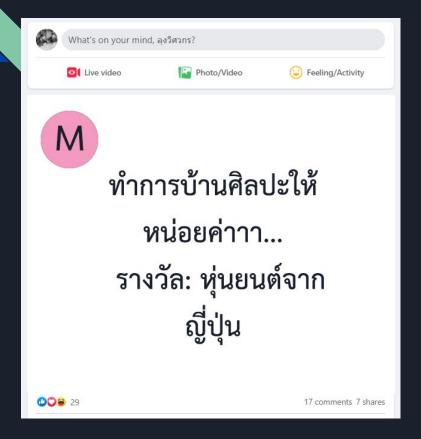
Git & GitHub ฉบับนิทาน EP.1



้มาฟังนิทานเรื่องนี้กันจ้าา...



กาลครั้งหนึ่งปี ค.ศ. 2050 หลังจากมนุษย์เริ่มไปตั้งรกรากบน ดาวอังคาร มีเด็กๆกำลังวิ่งเล่น ณ โรงเรียนแห่งหนึ่ง "เด็กหญิง มารุโกะจัง" ลูกครึ่งญี่ปุ่นเป็นนักเรียนใหม่ของโรงเรียนแห่งนี้ เธอ พึ่งเข้ามาเรียนในเทอมที่ 2 เพราะพึ่งกลับมาจากญี่ปุ่น เพื่อนๆได้ เรียนกันไป 1 เทอมแล้ว มารูโกะจังเป็นนักเรียนที่เก่งทุกวิชา ยกเว้นวิชาศิลปะ



้เนื่องจากมารูโกะจังเรียน ช้ากว่าเพื่อนๆ และวาดรูปไม่ ้เก่ง จึงโพสข้อความตามนี้ "こんにちは まるこです(คน ็นิจิว่ะ มะรุโกะเดส).. ทำการบ้านศิลปะให้ หน่อยค่าาา... รางวัล: หุ่นยนต์ จากญี่ปุ่นใครสนใจทักมา ็เลยจ้าาาา"



ทันใดนั้น "เด็กชาย ต.เต่า" ซึ่งเป็น friend กับมารูโกะจังแล้ว เห็น โพสดังกล่าวจึงทักมารูโกะจังไป แล้วบอกว่า เธอจะส่งงานพรุ่งนี้ ฉันทำไม่ทันหรอก ฉันขอตัวช่วย ให้ "เด็กชาย ป.ปลา" มาช่วย ด้วย เพื่อความรวดเร็ว....โอเคจ้าา





โรงเรียนเลิก ทั้ง 2 คนจึงรีบเดินทางกลับบ้าน ระหว่างที่เด็กชาย ต.เต่าและเด็กชายป.ปลากำลังเดินกลับ เดินผ่านบ้านที่กำลัง ก่อสร้างอยู่ เห็นลุงคนหนึ่งกำลังทำอะไรบางอย่าง ?!?!



ลุงคนนั้นคือใคร? "ลุงวิศวกร" ไงจะใครล่ะ เพี้ยนไม่เหมือนชาว บ้านแบบนี้ 555 ... เจ้าหนุ่ม! สนใจเครื่องมือตัวนี้ไหม? รู้ไหมมัน คืออะไร?



เขาเรียกเครื่องมือตัวนี้ว่าว่า "Git" (ชื่อเหมือนกิตติที่ โรงเรียนเลยลุง 55) ซึ่ง Git เป็นซอฟแวร์ประเภท Version Control System (VCS) เจ้า git ตัวนี้ลุงใช้ใน การจัดการ version ของไฟล์งาน และไฟล์โค้ดโปรแกรม ของลุง ปกติลุงจะมีทีมงานเขียน 3 คนรวมลุง แบ่งกันทำ คนละ 10 ไฟล์ รวมทั้งหมด 30 ไฟล์ แต่บางครั้ง นาย A ์ต้องแก้ไขงานของนาย B บางครั้งลุงก็ต้องแก้ไขงานของ ็นาย A นาย A work from home จากอเมริกา , นาย B อยู่ ญี่ปุ่น ส่วนลุงอยู่บนดาวอังคารกับพวกเธอนี่แหละการ จะรวมไฟล์ทั้งหมดถ้าไม่มีตัวช่วย จะปวดหัวมาก



ลุงจะใช้เจ้าตัว git คอยช่วยเหลือในการจัดการเรื่องไฟล์งาน git สามารถติดตามไฟล์งานได้อย่างละเอียด แบ่งเป็น เวอร์ชั่นการเขียนได้ (commit) ย้อนกลับไปใช้เวอร์ชั่นก่อน หน้าได้ แยกไฟล์ออกไปทดลองเป็นเหมือนโปรเจคย่อยได้ (branch) เพื่อไม่ให้กระทบกับเวอร์ชั่นล่าสุด หากจะรวมไฟล์ (merge) ก็สามารถรวมได้ รวมไปถึงสามารถสั่งอัพโห^ลด<u>ไฟล์</u> <mark>(push)</mark> ขึ้น GitHub (ลุงจะอธิบายใน EP.2) ได้ หรือจะดึง ข้อมูลที่อัพเดตล่าสุดจาก GitHub มารวมกับไฟล์ล่าสุดใน ้เครื่อง (pull)ก็ทำได้เช่นกัน รวมไปถึงการดาวโหลดโปรเจค ้อื่นๆ (clone) ไปจนตรวจสอบความแตกต่างระหว่างไฟล์ก่อน หน้ากับไฟล์ล่าสุด (diff) และความสามารถอื่นๆอีกมากมายที่ ็จะช่วยให้เราไม่ต้องเจอกับปัญหาเหล่านี้.....

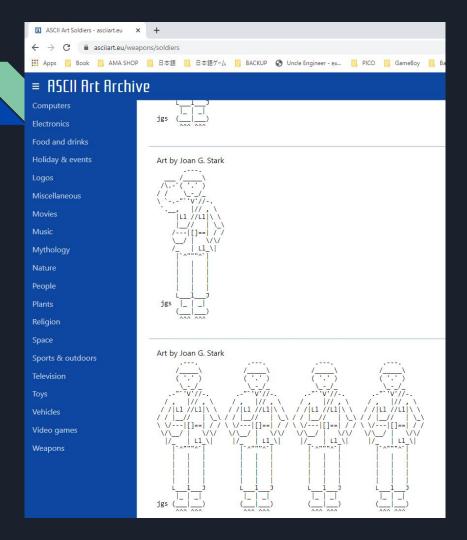
เคยเจอไหม? เวอร์ชั่นของงานที่เคยทำมีเยอะมาก 55

- File งานของลง V.1
- File งานของลง V.2
- File งานของลุง V.3
- File งานของลุง V.3 Final
- File งานของลง V.3 Final 1
- File งานของลุง V.3 Final 2 จากนาย A
- File งานของลุง V.3 Final 3 จากนาย B (แก้แล้วลุง)
- File งานของลุง V.3 Final 3 จากนาย B (ลุงพึ่งคอมเมนท์ แก้ใหม่ด่วน!)

ปวดหัวไหม? เจอหลายเวอร์ชั่นขนาดนี้ จะรวมอันไหนก่อนดี 555555555555555555

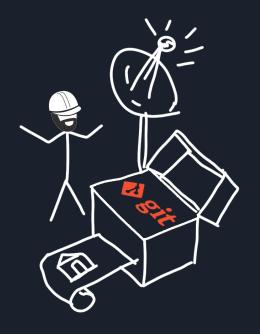
" ใช้ git แล้วชีวิตจะดีขึ้น "

ไอสไตน์ไม่ได้เข้าฝันมาบอกลุง



พวกเรารับงานศิลปะจาก มารูโกะจังมา ครูให้ทำ ASCII Art พวกเราจะใช้ วาดรูปแบบนี้ได้ไหม ลุง?

ได้สิ!...มาเลยเดี่ยวลุง สอน! ลุงจะบอก พวกนาย ต้องทำตามน่ะ! ครับลุง!



มาดูวิธีใช้งานเจ้า git กันเลยจ้าา

การติดตั้ง git

แวะไปบ้านหลังนี้เลย

https://git-scm.com

เขาแจก git ให้ใช้งานได้ฟรีๆจ้าาา





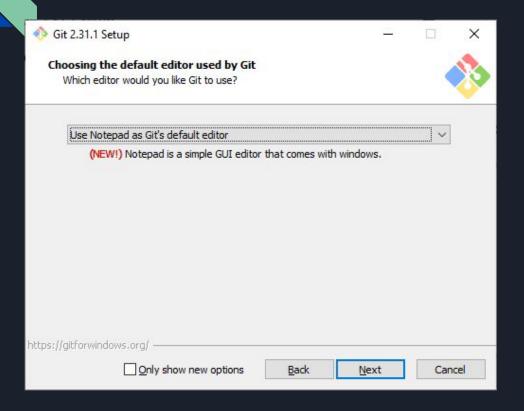
เข้าไปแล้วหาภาพนี้ให้เจอ แล้วคลิ กที่ Download 2.31.1 for Windows ได้เลย



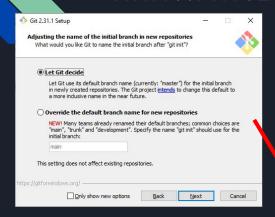
Next >

Which components should be installed? Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue. Additional icons On the Desktop Windows Explorer integration Git Bash Here Git GUI Here Git LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Use a TrueType font in all console windows				
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue. Additional icons On the Desktop Windows Explorer integration Git Bash Here Git GUI Here Git LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Use a TrueType font in all console windows	elect Components			
install. Click Next when you are ready to continue. Additional icons On the Desktop Windows Explorer integration Git Bash Here Git GUI Here Git LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Use a TrueType font in all console windows	Which components should be installed?			
Additional icons On the Desktop Windows Explorer integration Git Bash Here Git GUI Here Git LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Use a TrueType font in all console windows	Select the components you want to install; clear	r the compon	ents you do no	ot want to
☐ On the Desktop ☐ Windows Explorer integration ☐ Git Bash Here ☐ Git GUI Here ☐ Git LFS (Large File Support) ☐ Associate .git* configuration files with the default text editor ☐ Associate .sh files to be run with Bash ☐ Use a TrueType font in all console windows	install. Click Next when you are ready to continu	ue.		
Windows Explorer integration Git Bash Here Git GUI Here Git LFS (Large File Support) Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Use a TrueType font in all console windows	Additional icons			
☐ Git Bash Here ☐ Git GUI Here ☐ Git LFS (Large File Support) ☐ Associate .git* configuration files with the default text editor ☐ Associate .sh files to be run with Bash ☐ Use a TrueType font in all console windows	On the Desktop			
☐ Git GUI Here ☐ Git LFS (Large File Support) ☐ Associate .git* configuration files with the default text editor ☐ Associate .sh files to be run with Bash ☐ Use a TrueType font in all console windows	✓ Windows Explorer integration			
☐ Git LFS (Large File Support) ☐ Associate .git* configuration files with the default text editor ☐ Associate .sh files to be run with Bash ☐ Use a TrueType font in all console windows	- ☑ Git Bash Here			
Associate .git* configuration files with the default text editor Associate .sh files to be run with Bash Use a TrueType font in all console windows	☑ Git GUI Here			
Associate .sh files to be run with Bash Use a TrueType font in all console windows	Git LFS (Large File Support)			
Use a TrueType font in all console windows	Associate .git* configuration files with the	default text o	editor	
	Associate .sh files to be run with Bash			
Check daily for Git for Windows updates	Check daily for Git for Windows updates			
		disk space.		
Current selection requires at least 259.5 MB of disk space. /gitforwindows.org/	//gitforwindows.org/		454	202

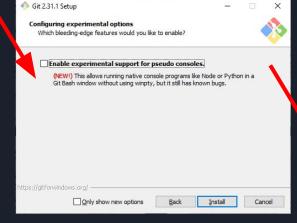
Next >>

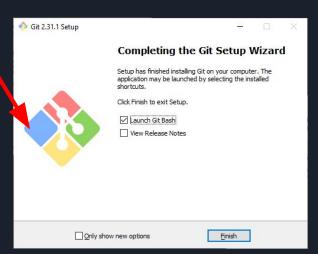


จุดนี้เขาให้เลือก
Editor > เลือกตัวที่
ตัวเองชอบเลย ถ้า
ไม่มีก็ใช้ Notepad
แล้วกด Next



Next เรื่อยๆ จนเจอ คำว่า Install ก็ติด ตั้งตามนั้น





ติดตั้งสำหรับ Mac



เข้าไปแล้วหาภาพนี้ให้เจอ แล้วคลิ กที่ Download 2.31.0 for Mac ได้เลย

ติดตั้งสำหรับ Mac

Homebrew

Install homebrew if you don't already have it, then:

\$ brew install git

เปิด Terminal (จอสีดำ) แล้วพิมพ์

brew install git

จบ!พร้อมใช้ แค่นี้ 555

ถ้าไม่มีให้กดคำว่า homebrew สีฟ้า จะเจอวิธีติดตั้ง homebrew

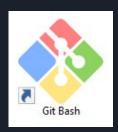
จะมีคำสั่งนี้ให้พิมพ์...

/bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

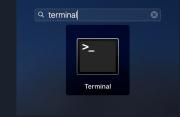
้ถ้าพร้อมแล้วเปิดโปรแกรมรอไว้เลย....

โปรแกรมที่จะเราจะใช้ สั่งงานจะทำผ่าน command line interface (CLI) ทั้งหมด

Windows ให้หาโปรแกรมชื่อ Git Bash ในเมนู Start



Mac ใช้ Terminal ได้เลยจำาว



ระหว่างนี้ทุกคนจะได้เรียนรู้

bash command

บางส่วนไปในตัว

ต่อไปเข้าไป Desktop เพื่อเตรียมสร้างโฟลเดอร์

```
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~ (master)

$ pwd
/c/Users/Uncle Engineer

Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~ (master)

$ cd Desktop/

Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop (master)

$ ____
```

Desktop — -zsh — 58×8

Last login: Mon Apr 19 22:55:12 on ttys000

uncleengineer@Uncles-MacBook-Pro ~ % pwd

/Users/uncleengineer

uncleengineer@Uncles-MacBook-Pro ~ % cd Desktop/
uncleengineer@Uncles-MacBook-Pro Desktop %

pwd

เพื่อเช็คว่าตัวเองอยู่ที่ไหน

cd Desktop\

เพื่อเข้าไปหา Desktop

*** หลังจากนี้ลุงขอทำตัวอย่างใน windows ส่วน mac ก็ใช้คำสั่งเหมือนกัน

สร้างโฟลเดอร์ (mkdir) , สร้างไฟล์ใหม่ (touch)

```
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop (master)

$ mkdir asciiart

Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop (master)

$ cd asciiart/

Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)

$ touch marukowork.py

Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)

$ ____
```

mkdir asciiart

เพื่อสร้างโฟลเดอร์ชื่อว่า asciiart

cd asciiart/

้เพื่อเข้าไปหาโฟลเดอร์ asciiart

touch marukowork.py

เพื่อสร้างไฟล์ marukowork.py ขึ้นมาเป็นไฟล์งานหลัก

เพิ่มข้อความใส่เข้าไปในไฟล์ marukowork.py

```
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ echo "print('Hello Maruko')" >> marukowork.py

Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ cat marukowork.py
print('Hello Maruko')
```

พิมพ์ตามด้านซ้าย คำสั่ง echo ในที่นี้ถ้ามี เครื่องหมาย >> แล้วตาม ด้วยชื่อไฟล์ จะเป็นการ เพิ่มข้อความเข้าไปใน ไฟล์นั้น

ใช้ cat เพื่อแสดงสิ่งที่อยู่ ในไฟล์นั้น

ลุงจะเริ่มคำสั่ง git แล้วนะจ๊ะ

Basic git command

ขั้นแรกมาตั้งค่า git config กันก่อน

```
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git config --global user.name "Dekchai Tortao"

Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git config --global user.email dekchai.tortao@gmail.com

Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$
```

ใช้คำสั่งตามนี้ บรรทัดแรก ตั้งค่า ชื่อผู้แก้ไข ใช้ชื่อ แทนตัวเองแทน "Dekchai Tortao" ได้เลย

บรรทัดต่อมาคือตั้งค่าอี เมลล์ ใช้อีเมลล์ตัวเองแทน ได้เลยจ้าาา

git init ตั้งค่าโฟลเดอร์ให้เป็น local git repo (ฯ)

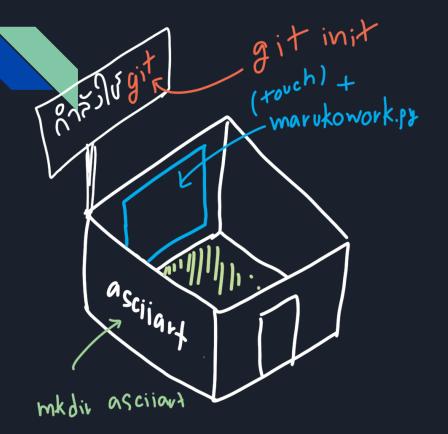
```
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git init
Initialized empty Git repository in C:/Users/Uncle Engineer/Desktop/asciiart/.git/
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ ls -a
./ ../ .git/ marukowork.py
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ __
```

git init

คำสั่งตั้งค่าให้โฟลเดอร์ที่เรา กำลังทำงานเป็น local git repository (repo)

ls -a

คำสั่งแอลเอสขีดเอ เป็น
คำสั่ง list โชว์ว่าใน
โฟลเดอร์ ณ ตอนนี้มี
ไฟล์หรือโฟลเดอร์ไหนบ้าง
ทั้งที่ซ่อนและไม่ซ่อน



ลุงเล่านิทานต่อ.....

ตอนนี้เด็กทั้งสองพร้อมจะทำงานให้มารู
โกะจังแล้ว ทั้งสองตัดสินใจว่าจะสร้างห้อง
เฉพาะขึ้นมาห้องหนึ่ง (สร้างโฟลเดอร์
asciiart) สำหรับทำงานให้มารูโกะจังโดย
เฉพาะ โดยในห้องจะติดกระดาษแผ่นใหญ่
ไว้ทำงานศิลปะให้มารูโกะจัง (ไฟล์
marukowork.py)

คำสั่ง git add เพื่อติดตามสถานะของไฟล์

```
- □ ×
MINGW64:/c/Users/Uncle Engineer/Desktop/asciiart
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git add marukowork.py
warning: LF will be replaced by CRLF in marukowork.py.
The file will have its original line endings in your working directory
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file: marukowork.py
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$
```

git add marukowork.py

คำสั่ง add คือคำสั่งเพื่อบอก git ว่าให้ติดตามการเปลี่ยนแปลง ของไฟล์ marukowork.py ถ้า ไฟล์มีการแก้ไขเมื่อไร git จะแจ้ง เราผ่านคำสั่ง status ต่อมาทันที (ขณะนี้เขาเรียกสถานะว่า stated ไว้ลุงจะอธิบายเพิ่มให้จ้าา)

git status

คำสั่ง status ใช้สำหรับเช็ค สถานะไฟล์ว่าตอนนี้เกิดอะไรขึ้น กับไฟล์

git add marukonork.py

ลุงเล่านิทานต่อ.....

เมื่อสั่ง git add marukowork.py เจ้า git จะส่งกล้องมาตรวจจับทันทีว่าให้ติดตามไฟล์นี้เลย แล้วส่งไปเก็บไว้บนจอของ staged

ต่อมาเมื่อสั่ง git status ก็จะเห็นว่าระบบ เช็คได้ว่ามีไฟล์ใหม่ (new) เกิดขึ้นชื่อ ไฟล์ว่า marukowork.py

คำสั่ง git commit เพื่อยืนยันการบันทึกเวอร์ชั่น

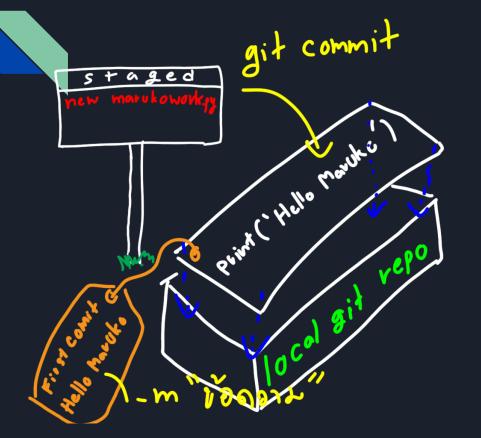
```
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git commit -m "First Commit Hello Maruko"
[master (root-commit) Oced80c] First Commit Hello Maruko
1 file changed, 1 insertion(+)
create mode 100644 marukowork.py
Uncle Engineer@DESKTOP-OT644K4 MINGW64 ~/Desktop/ascijart (master)
$ git log
commit 0ced80c7b499b58846ab4aefdda1ff86bbdf8fb7 (HEAD -> master)
Author: Dekchai Tortao <dekchai.tortao@gmail.com>
Date:
       Mon Apr 19 23:53:11 2021 +0700
   First Commit Hello Maruko
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/ascijart (master)
$_
```

git commit -m "First Commit Hello Maruko"

คำสั่ง commit เป็นคำสั่งเพื่อบอก git ว่าให้ยืนยันไฟล์ที่สถานะ staged แล้วส่งเข้าไปเก็บไว้ใน local git repo หลัก เพื่อบันทึกว่า เวอร์ชั่นนี้ เขียนไปถึงไหน ใครเป็น คนอัพเดต อัพเดตตอนไหน มี ข้อความสั้นๆว่าอะไร

git log

คำสั่ง log เพื่อเช็คว่าตอนนี้มีการ commit มากี่ครั้งแล้ว



ลุงเล่านิทานต่อ.....

เมื่อ git commit เปรียบเสมือน git สั่งแคปเจอร์ไฟล์และโฟลเด อร์ทั้งหมดที่มาจากไฟล์ที่ staged แล้วมาแล้ว (ผ่านการ add แล้ว) แล้ว ติดแท็กข้อความ -m ไว้ว่า First Commit Hello Maruko เพื่อใช้ใน การติดตาม version ของไฟล์ การ ีย้อนกลับมาแก้ไข แยก branch ออก ไป (อ่านต่อ ep หน้า)

ต่อไปจะทดลองเพิ่มข้อความใส่อีกบรรทัดเข้าไปในไฟล์ marukowork.py

```
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ echo "print('-----Welcome to ASCII ART------')" >> marukowork.py
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ cat marukowork.py
print('Hello Maruko')
print('-----Welcome to ASCII ART-----')
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ __
```

เพิ่มอีกบรรทัดใช้คำสั่ง
echo "ข้อความ" >>
ชื่อไฟล์ เหมือนก่อนหน้า
จะเป็นการเพิ่มข้อความ
เข้าไปในไฟล์

ใช้ cat เพื่อดูไฟล์ว่า เขียนอะไรไว้บ้างเช่น เดิม

เมื่อเช็ค status อีกครั้ง จะเจอไฟล์ที่มีการแก้ไข

หลังจากเพิ่มข้อความอีก บรรทัดแล้ว เช็ค status ไฟล์ จะเห็นว่ามีคำว่า modified ขึ้นมา แสดงว่ามี การแก้ไขไฟล์นี้แล้ว

หลังจากแก้ไฟล์ ให้ git add , git commit

```
Uncle Engineer@DESKTOP-OT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git add marukowork.py
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git commit -m "add Welcome"
[master 21ef426] add Welcome
 1 file changed, 1 insertion(+)
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git log
commit 21ef426fb9fde242b144dcc79674062727e1bff7 (HEAD -> master)
Author: Dekchai Tortao <dekchai.tortao@gmail.com>
Date:
        Tue Apr 20 01:56:09 2021 +0700
    add Welcome
commit 0ced80c7b499b58846ab4aefdda1ff86bbdf8fb7
Author: Dekchai Tortao <dekchai.tortao@gmail.com>
        Mon Apr 19 23:53:11 2021 +0700
Date:
    First Commit Hello Maruko
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$
```

git add marukowork.py เพื่อ อัพเดตการแก้ไขไปเก็บไว้ใน staged จากนั้น git commit -m "add Welcome" เพื่อส่งไปเก็บ ไว้ใน local git repo เป็นอันจบ กระบวนการ

เมื่อ git log จะสังเกตเห็นว่ามี การอัพเดตข้อมูลการแก้ไขเพิ่ม ขึ้นมาอีก 1 รายการ รวมเป็น 2 รายการ

ทดลองสร้างไฟล์ใหม่

ทดลองสร้างไฟล์ใหม่ touch art-cat.py แล้ว copy ข้อความรูปแมว ascii art จากเว็บไซต์ https://www.asciiart.eu/animals/cats จากนั้นเปิดไฟล์ art-cat.py ด้วย Notepad หรือ Editor ที่ถนัด จากนั้นเซฟไฟล์ให้ เรียบร้อย ...(ไว้ลุงจะมาสอนใช้ editor ชื่อ nano ตอนหน้า)

หลังจากแก้ไขไฟล์ หากไม่มีการ git add จะเกิดอะไรขึ้น?

```
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
 cat art-cat.pv
cat = '''
print(cat)
Uncle Engineer@DESKTOP-OT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        art-cat.py
nothing added to commit but untracked files present (use "git add" to track)
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
```

ทดลอง cat art-cat.py เพื่อดูว่า มีข้อความที่แก้ไขหรือไม่?

จากนั้น git status เพื่อตรวจ สอบสถานะไฟล์ ตอนนี้สถานะ ของไฟล์ art-cat.py ซึ่งยังไม่ เคย git add จะมีสถานะว่า "untracted" ซึ่งเป็นสถานะที่ บอกถึงไฟล์ที่ไม่ได้ถูกติดตาม

git add. git log --oneline

```
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git add .
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git commit -m "add art-cat.py"
[master 67eb09d] add art-cat.py
1 file changed, 7 insertions(+)
 create mode 100644 art-cat.py
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
$ git log --oneline
67eb09d (HEAD -> master) add art-cat.py
f2f6ead add Welcome
Oced80c First Commit Hello Maruko
Uncle Engineer@DESKTOP-QT644K4 MINGW64 ~/Desktop/asciiart (master)
```

git add . (กิทแอดเคาะวรรคแล้ว จุดหนึ่งครั้ง) เป็นการเพิ่ม ไฟล์หรือโฟลเดอร์ที่เกี่ยวข้อง ทั้งหมด ใช้ในกรณีที่ไฟล์มีมาก ขึ้น ไม่ต้อง add ทีละไฟล์

git commit -m "add art-cat.py" เพื่อส่งไปเก็บไว้ใน local git repo บันทึกว่า เพิ่มไฟล์ art-cat.py

git log --oneline คำสั่งนี้ช่วย ให้เราดูแค่รหัสรายการและข้อ ความบันทึกในการ commit แต่ละครั้ง

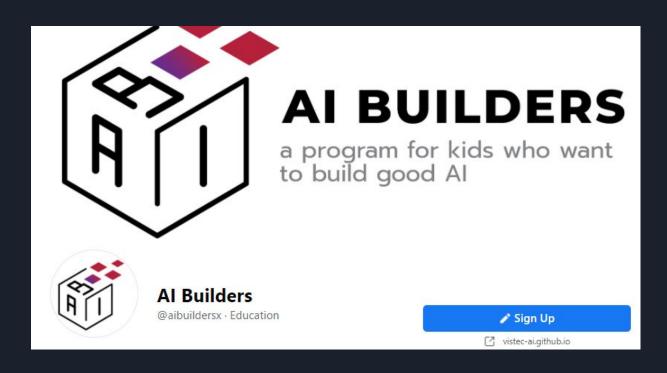
つづく

โปรดติดตามตอนต่อไปเรื่อง ...

- GitHub คืออะไร? สมัครอย่างไร?
- Push โปรเจคขึ้นไปบน GitHub ทำได้อย่างไร?
- อัพเดตข้อมูล หรือ pull ข้อมูลล่าสุดจาก GitHub มารวมกับโปร เจคปัจจุบันที่แก้ไข ทำได้อย่างไร?
- เด็กชายป.ปลา อยากจะแก้ไขไฟล์จะทำอย่างไร?
- ตรวจสอบความแตกต่างระหว่างไฟล์ล่าสุด และไฟล์ก่อนหน้า
- อื่นๆ

ใครอยากดูไลฟ์เจอกันในเพจ Al Builders

https://www.facebook.com/aibuildersx



์ติดตาม Event ลุงจะไลฟ์วันพฤหัสที่ 22 เม.ย. 2021 นี้เวลา 18.00 เป็นต้นไป





https://fb.me/e/5vSfHCKD4

แต้งกิ้วที่อ่านจบ ว่าแต่ใครอ่านจบบ้าง 5555

***** คำเตือน บทความนี้ หากไม่ลงมือทำ อาจจะงงว่ามันคืออะไร? ๕๕๕๕