



Software Park Thailand
</Code Camp>

Git



Coach Bhornumnard Wanasrisun

Outline

- Git ?
- Git host มีใครบ้าง
- Git register and create repository
- Git state
- Git command
- Git flow
- Git flow with team

Learning Outcome

เพื่อให้ นักเรียนมีความรู้ความเข้าใจในการใช้ Git และสามารถใช้งาน Git ได้อย่างมีประสิทธิภาพ เข้าใจหลักการทำงานของ Git และ สถานะต่างๆ และสามารถทำงานร่วมกับผู้อื่นได้

What is Git?



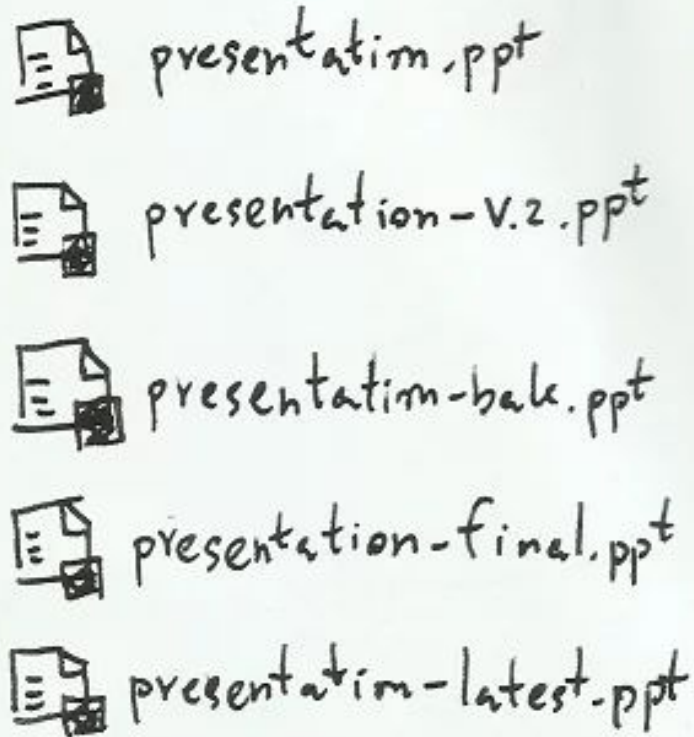
Software Park Thailand
Code Camp

Git คือ Version Control แบบ Distributed ตัวหนึ่ง เป็นระบบที่ใช้จัดเก็บและควบคุมการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ชนิดใดก็ได้ ไม่ว่าจะเป็น Text File หรือ Binary File (จากนี้จะขอเรียก Text File หรือ Binary File รวมกันว่า Source Code)

การจัดการ source code ในอดีต

ในอดีตการจัดการงานต่าง ๆ นั้นยังไม่มีระบบการจัดการที่ไฟล์ที่มีประสิทธิภาพมากนัก ข้อเสียต่าง ๆ ที่สามารถเห็นได้ชัดจากการจัดการไฟล์ คือ มีความซับซ้อน มีความซ้ำซ้อน การเก็บข้อมูลใน local workstation

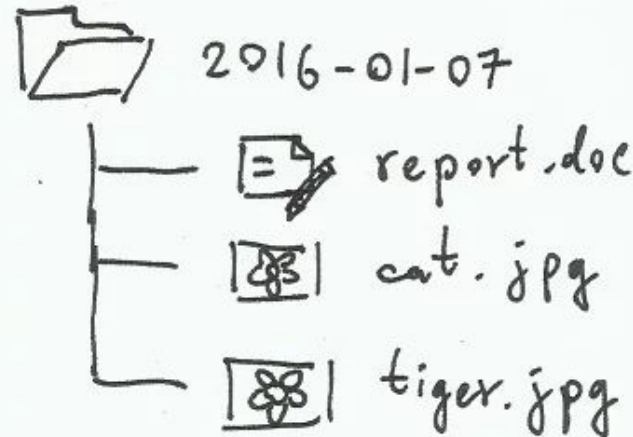
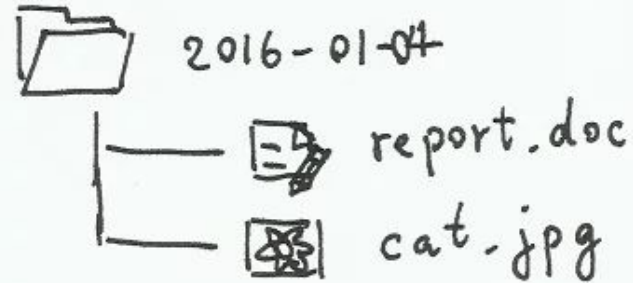
ยุกระบบจัดการชอริส แบบ เวอร์ชัน หรือ แบบดั้งเดิม



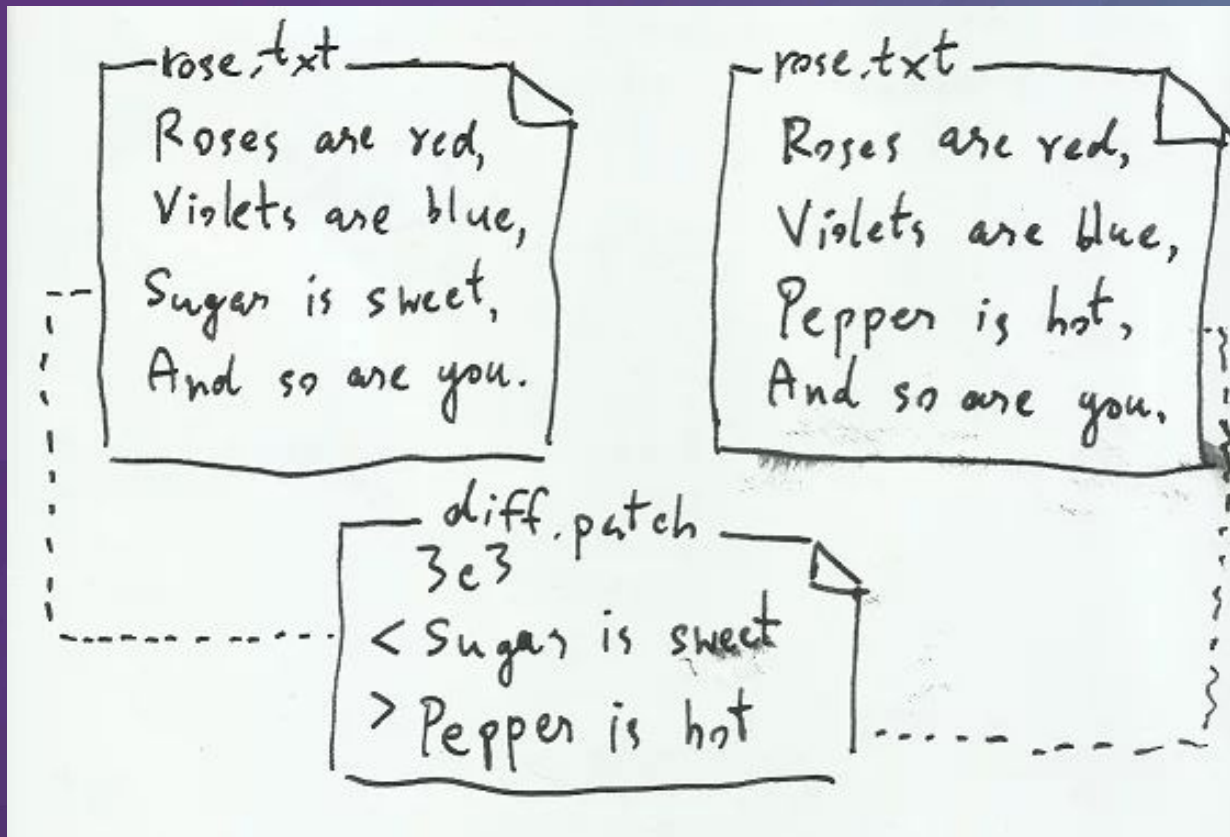
A handwritten list of five presentation files, each preceded by a small icon of a document with a folded corner. The files are listed vertically:

- presentation.ppt
- presentation-v.2.ppt
- presentation-back.ppt
- presentation-final.ppt
- presentation-latest.ppt

ยุคระบบจัดการชอริสด้วย folder ไฟล์วันที่

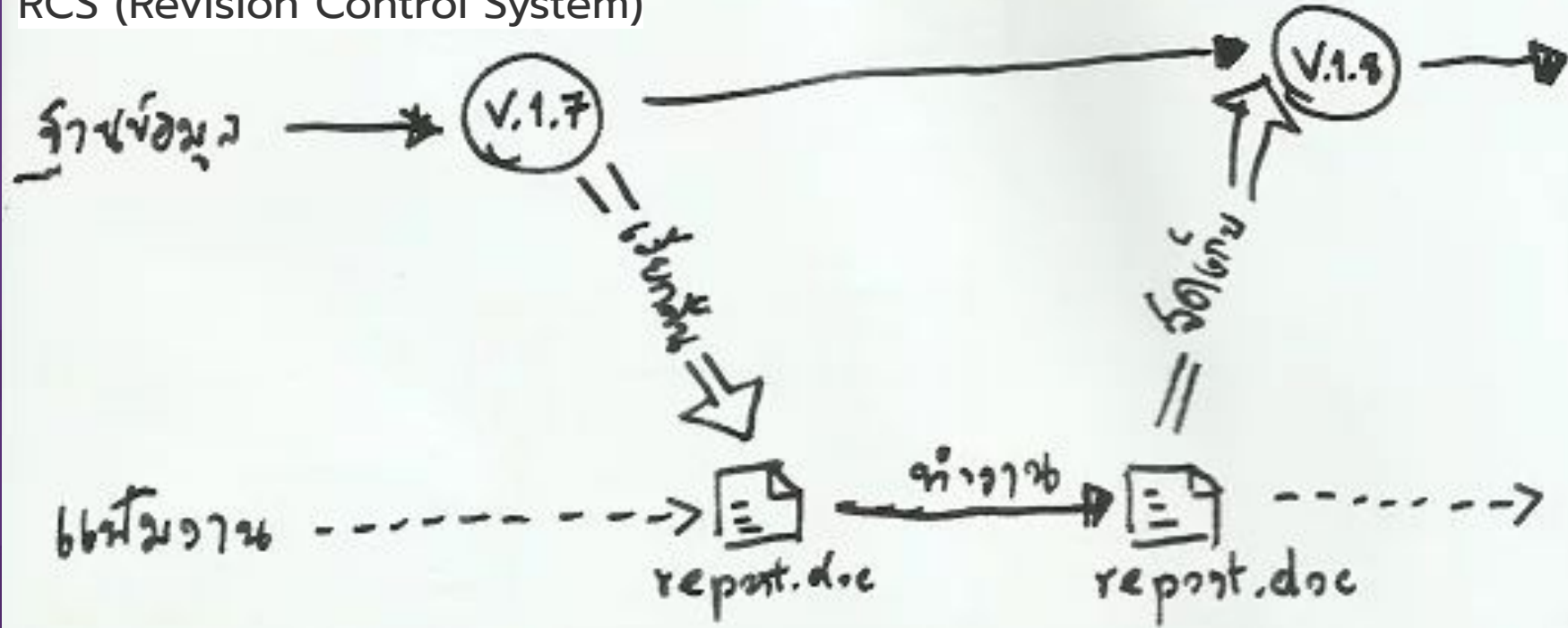


ยุกระบบจัดการชอริสด้วย patch(ส่วนที่แตกต่างกัน)

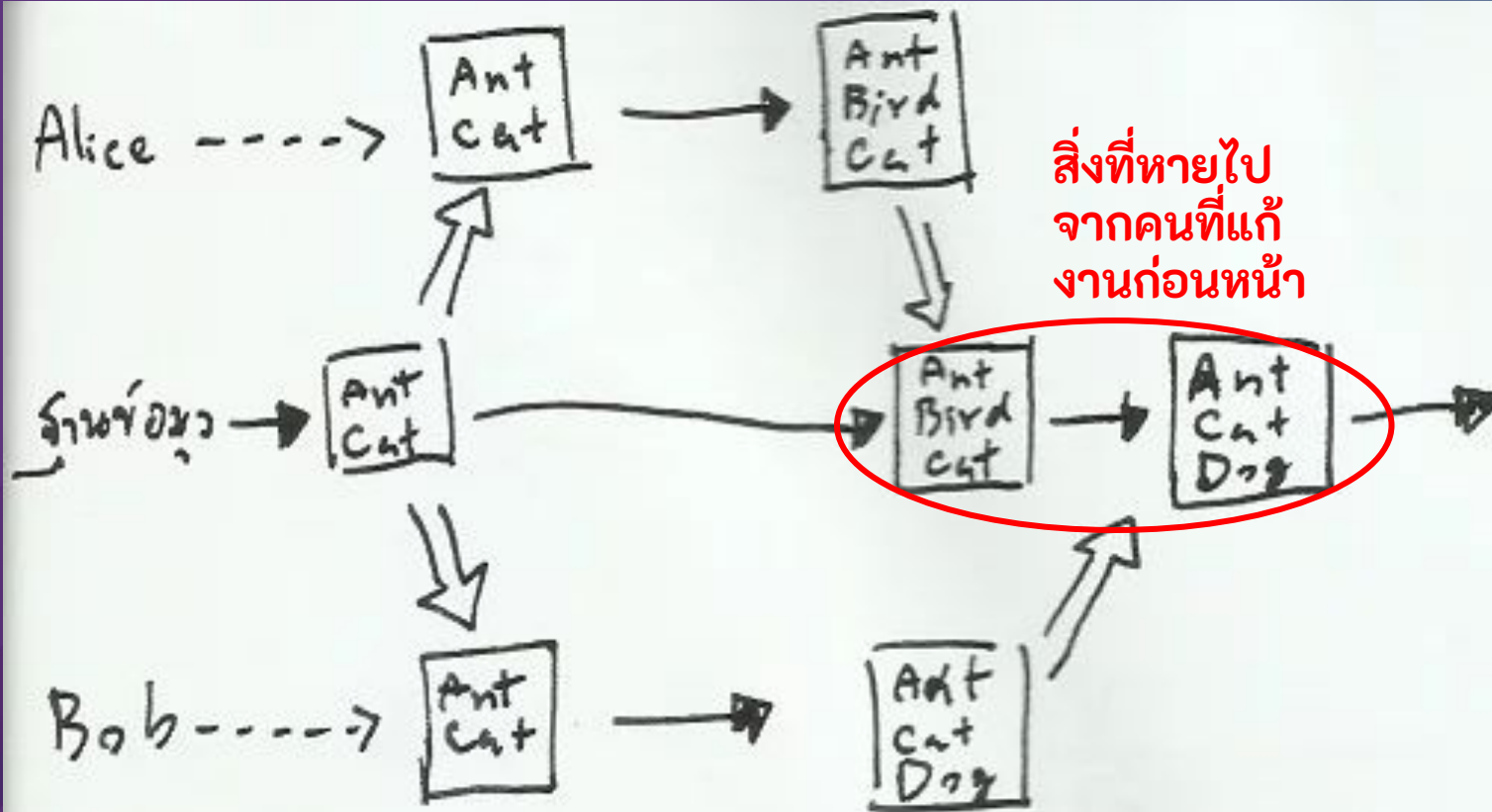


ยุคระบบจัดการซอร์สแบบเฉพาะที่ (Local Version Control System)

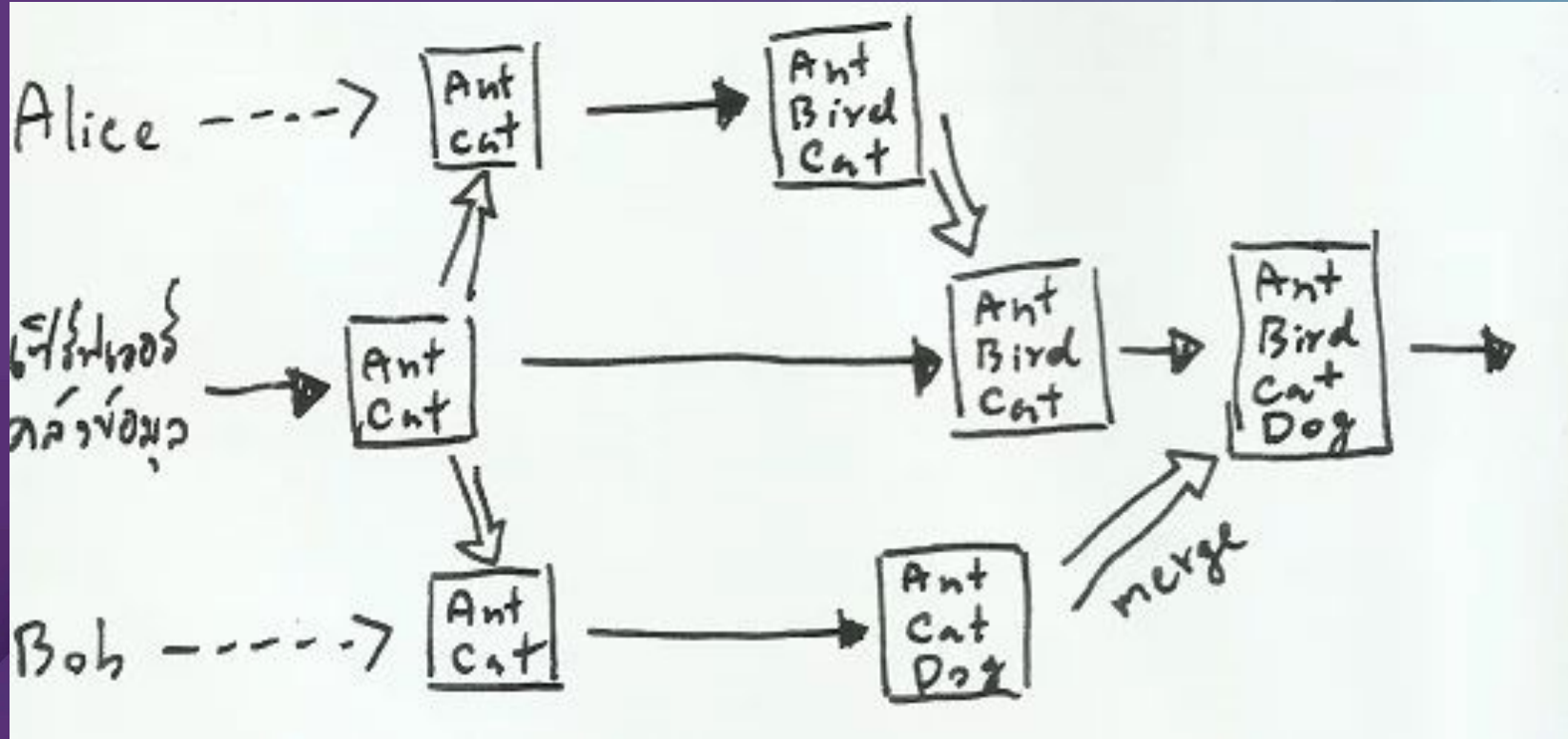
RCS (Revision Control System)



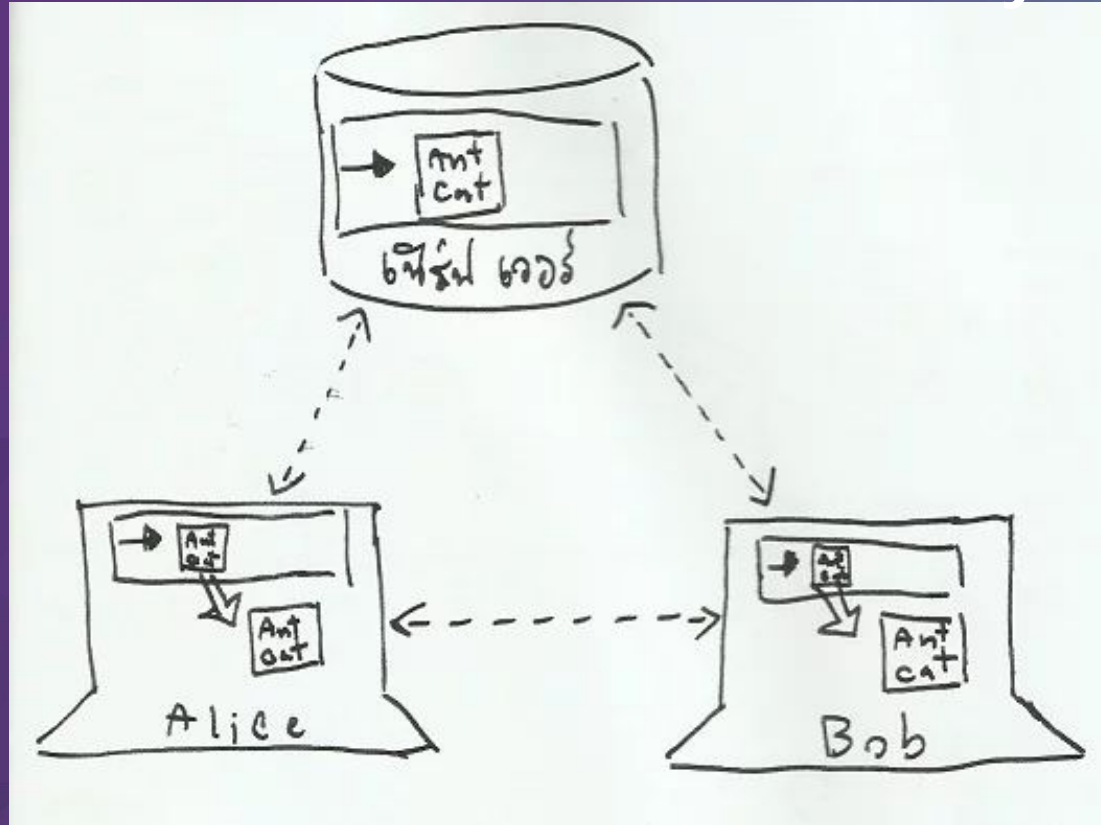
ปัญหาที่เกิดขึ้นจากการทำงานร่วมกันเป็นทีม



ยุกระบบจัดการซอร์สแบบรวมศูนย์ (Centralized Version Control System)



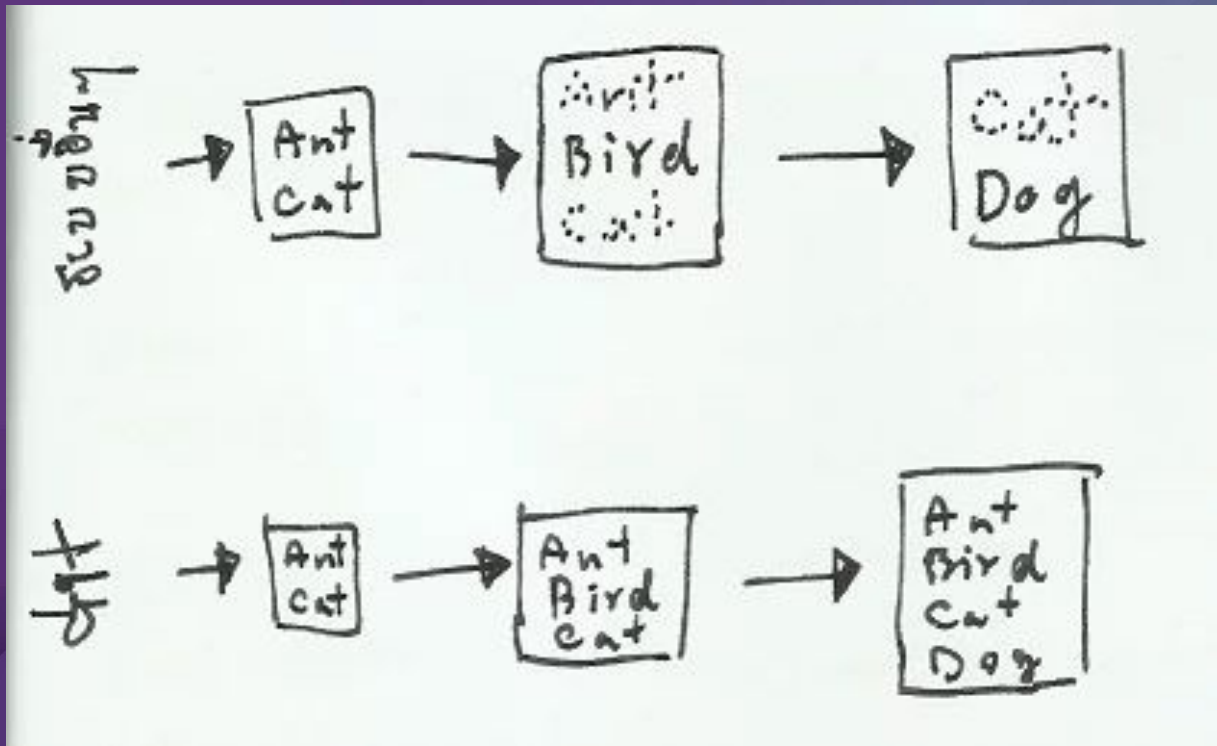
ยุกระบบจัดการซอร์สแบบกระจายศูนย์ (Distributed Version Control System)



สุดท้าย git ก็ถูกสร้างขึ้นโดยชุมชน linux



แนวคิดการมองภาพรุ่นของ Git เมื่อเทียบกับระบบอื่นๆ



[Ref git history](#)

Git host ใน ปัจจุบันมีใครบ้าง



Software Park Thailand
Code Camp
Software Park Thailand
</Code Camp>



GitHub



Bitbucket



GitLab



Azure DevOps



github register



Software Park Thailand
</Code Camp>

Pick a username

Your email


Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

git lab register

← → ↻ 🔒 https://gitlab.com/users/sign_in



GitLab.com offers free unlimited (private) repositories and unlimited collaborators.

- [Explore projects on GitLab.com](#) (no login needed)
- [More information about GitLab.com](#)
- [GitLab.com Support Forum](#)
- [GitLab Homepage](#)

By signing up for and by signing in to this service you accept our:

- [Privacy policy](#)
- [GitLab.com Terms.](#)

Username or email


Password


☐ Remember me [Forgot your password?](#)


Sign in


Don't have an account yet? [Register now](#)


Sign in with

 Google

 GitHub

 Twitter

 Bitbucket

 Salesforce

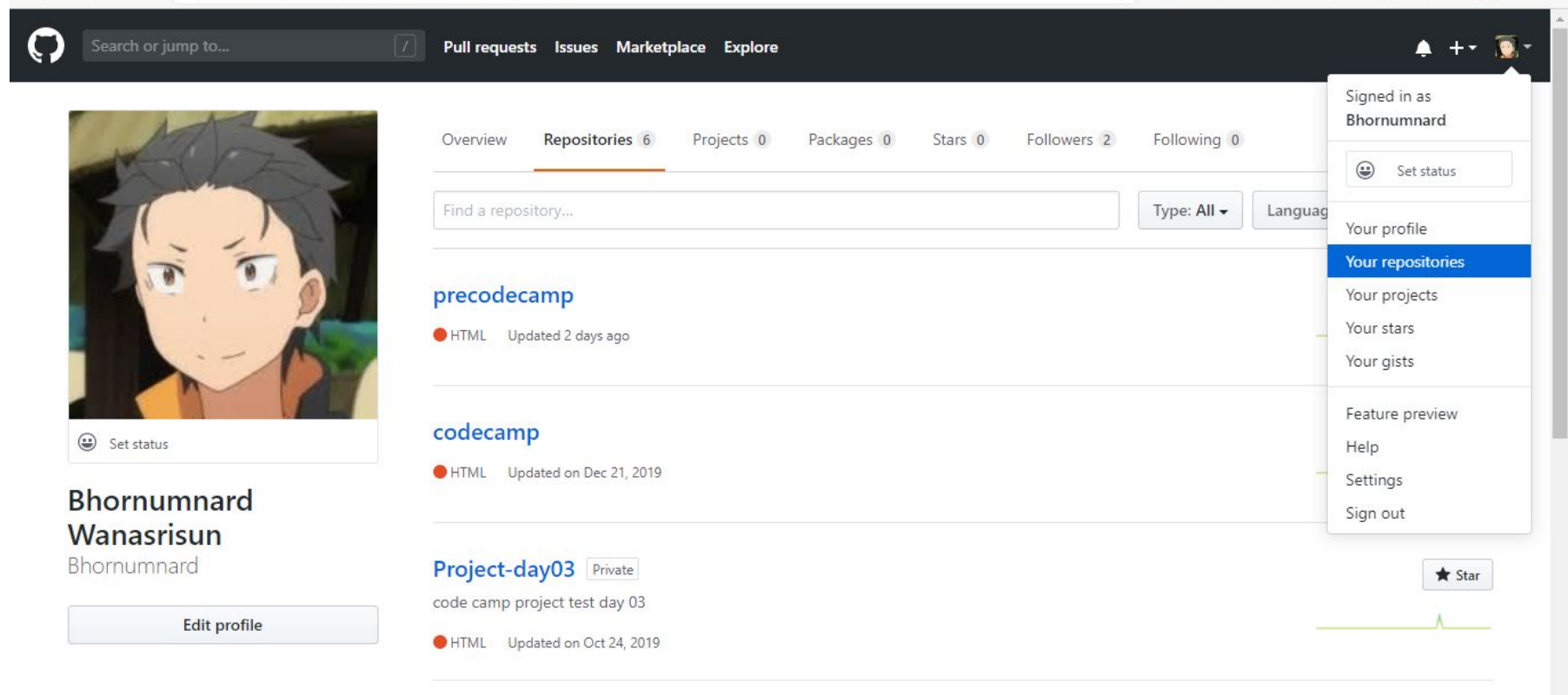
☐ Remember me

[git lab register with github](#)

Repository

repository คือ folder ที่ใช้สำหรับเก็บงาน project
ของเรา ซึ่งเราสามารถสร้าง branch (folder ย่อย)
เพื่อแยกไปเขียน code ก่อนที่เราจะเป็น
เวอร์ชันสมบูรณ์

github create repository



Search or jump to... / Pull requests Issues Marketplace Explore

Signed in as Bhornumard

Set status

Your profile

Your repositories

Your projects

Your stars

Your gists

Feature preview

Help

Settings

Sign out

Overview **Repositories 6** Projects 0 Packages 0 Stars 0 Followers 2 Following 0

Find a repository... Type: All Language

precodecamp

HTML Updated 2 days ago

codecamp

HTML Updated on Dec 21, 2019

Project-day03 Private

code camp project test day 03

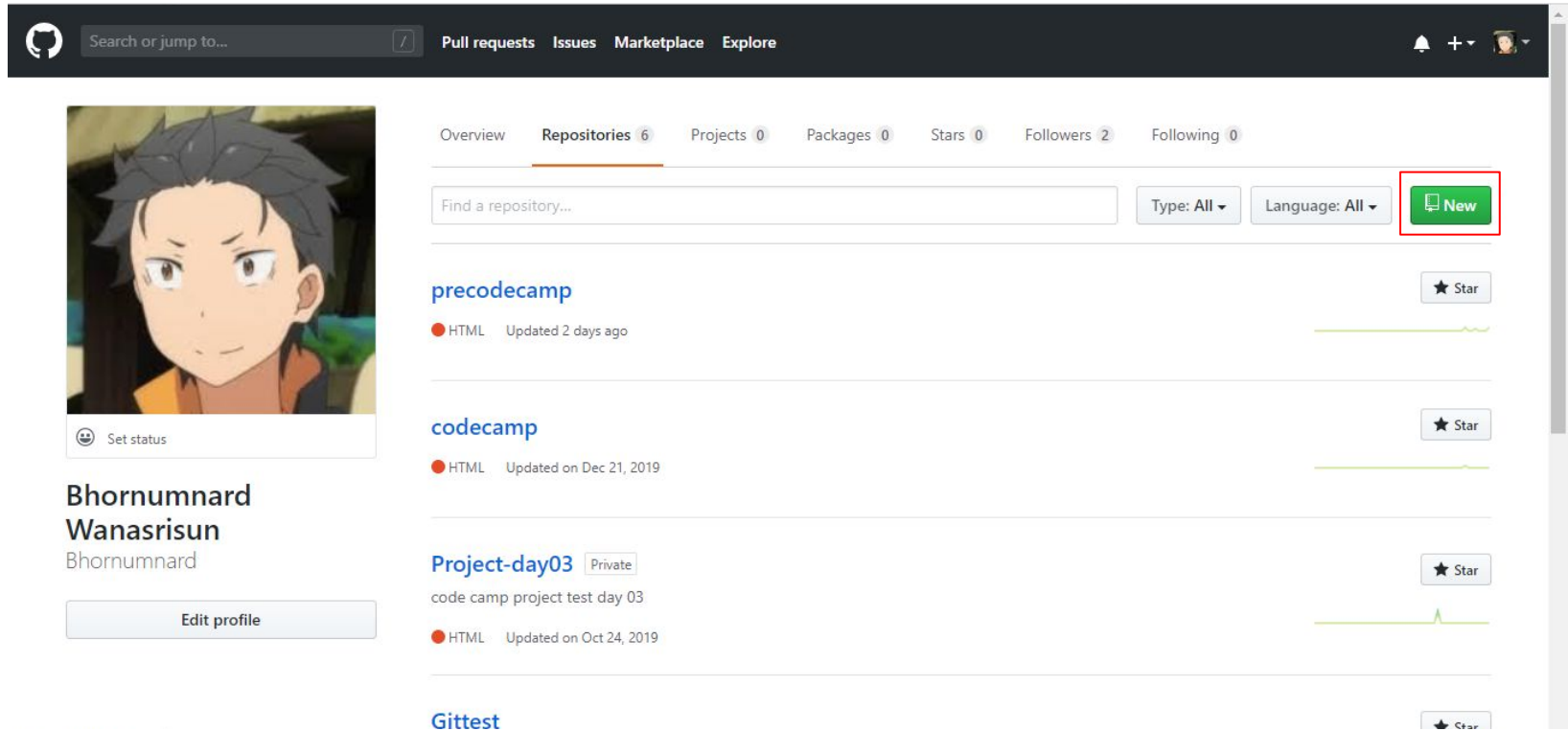
HTML Updated on Oct 24, 2019

★ Star

Bhornumard Wanasrisun
Bhornumard

Edit profile

github create repository



The screenshot shows the GitHub profile of a user named Bhornumnard Wanasrisun. The profile includes a profile picture of an anime-style character, a bio, and a list of repositories. The repositories listed are 'precodecamp', 'codecamp', and 'Project-day03'. The 'New' button in the repository filter section is highlighted with a red box.

Search or jump to... Pull requests Issues Marketplace Explore

Overview **Repositories 6** Projects 0 Packages 0 Stars 0 Followers 2 Following 0

Find a repository... Type: All Language: All **New**

precodecamp ★ Star
HTML Updated 2 days ago

codecamp ★ Star
HTML Updated on Dec 21, 2019

Project-day03 Private ★ Star
code camp project test day 03
HTML Updated on Oct 24, 2019

Gitttest ★ Star

github create repository

Owner



Repository name *

Great repository names are short and memorable. Need inspiration? How about [expert-adventure](#)?

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None ▼

Add a license: None ▼



Create repository

git bash (git command line)

https://git-scm.com/downloads

git --distributed-even-if-your-workflow-isnt

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads

Mac OS X Windows Linux/Unix

Latest source Release
2.25.0
Release Notes (2020-01-13)
Download 2.25.0 for Windows

Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools ([git-gui](#), [gitk](#)), but there are several third-party tools for users looking for a platform-specific experience.
[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
[View Logos →](#)

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

[gitbash download](#)

git bash display



Software Park Thailand
Code Camp



```
MINGW64:/c/Users/Pornumnard

Pornumnard@Dell-inspiron-3537 MINGW64 ~ (master)
$ |
```



Software Park Thailand
</Code Camp>

basic command

Basic command (should know)

- git init : สร้างไฟล์ .git เพื่อเก็บ state และ ความเปลี่ยนแปลง
- git clone : คัดลอก repo มาลงในเครื่องเรา
- git add : เพิ่ม state เพื่อเตรียม commit
- git commit : เตรียมข้อมูลการเปลี่ยนแปลง และ คำอธิบาย
- git status : เช็คสถานะของ state ณ ปัจจุบัน
- git merge : รวมไฟล์ ระหว่าง 2 branch
- git pull : download ข้อมูล จาก repo ลงเครื่อง
- git push : upload ข้อมูล จากเครื่องขึ้น repo

[สรุป git command pdf](#)



Software Park Thailand
</Code Camp>

Lab

New repository

- ให้สร้าง git repository ใหม่ ขึ้นมาว่า git test
- ทำการสร้าง folder ที่จะเก็บไฟล์ภายในเครื่อง
- คลิกขวาที่ folder ที่สร้างขึ้น มาใหม่และรัน git bash

- ทำการเชื่อม github กับ computer เรา
- สร้างไฟล์ Readme.md

markdown

markdown คือ markup language ชนิดหนึ่ง ที่เป็นการแสดงรูปแบบข้อความ ลำดับ และ สิ่งต่างๆ คล้ายๆกับ HTML markdown นิยมใช้ในการแสดงคำอธิบายต่างๆ เพราะว่ามียูสเอจในการจัดการข้อความที่ง่าย และ แสดงผลข้อมูลได้อย่างมีประสิทธิภาพ

Ref. [จุดประสงค์ในการสร้าง readme.md](#)

การเขียน markdown

#h1 = <h1> </h1>

##h2 = <h2> </h2>

###h3 = <h3> </h3>

####h4 = <h4> </h4>

#####h5 = <h5> </h5>

#####h6 = <h6> </h6>

2 spacebar enter =

การเขียน markdown

****content****, **__content__** = ** **

content, *_content_* = *<i> </i>*

******content******, ***___content___*** = ***<i> </i>***

__*content*__, *****_content_***** = ***<i> </i>***

> content = | content

>> content = nested blockquotes

การเขียน markdown

1. content
2. content
3. content
4. content

+ content
- content
*content
tab * content

1. content
2. content
3. content

- content
- content
- content
 - content

การเขียน markdown

![detail pic] (url) = picture

***, ---, ----- = <hr>

[content](url) = content



Software Park Thailand
</Code Camp>

Lab

Markdown

- ให้ลองเขียน markdown file
- สร้าง README.md
- ให้ใส่รายละเอียด repository หรือ ข้อมูลพื้นฐานต่างๆ

- ใช้คำสั่ง header
- สร้าง link
- สร้าง list
- สร้าง ตัวหนา ตัวเอียง
- สร้าง `<hr>`
- สร้าง **blockquotes**



Software Park Thailand
</Code Camp>





git state

git state

Local

working
directory

staging
area

localrepo

Remote

remote
repo



Software Park Thailand
</Code Camp>

git flow

Git flow

upload file (work directory => remote repro)

1. edit file
2. git add .
3. git commit -m "title" -m "description"
4. git push

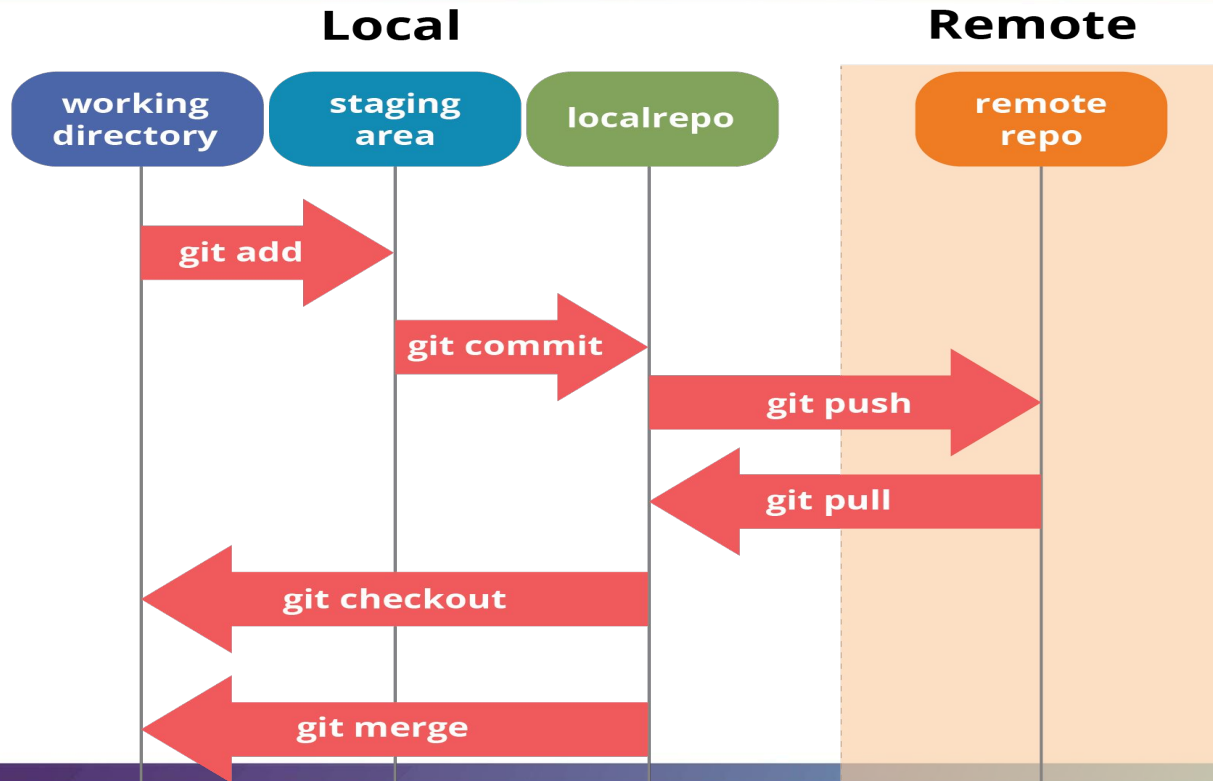
download file (remote repro => work directory)

1. git pull

Git flow



Software Park Thailand
Code Camp
Software Park Thailand
</Code Camp>



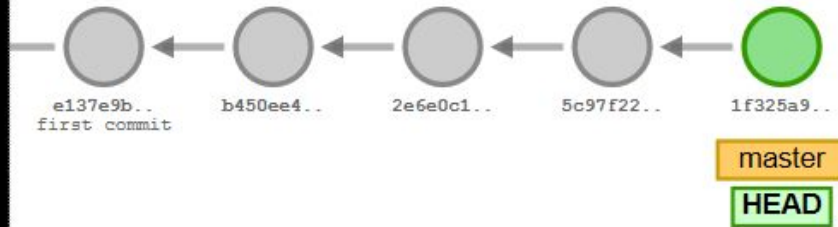
Git graph



Software Park Thailand
Software Park Thailand
</Code Camp>

```
Have fun!  
$ git commit  
$ git commit  
$ git commit  
$ git commit
```

HEAD: master





Software Park Thailand
</Code Camp>

Lab

git flow

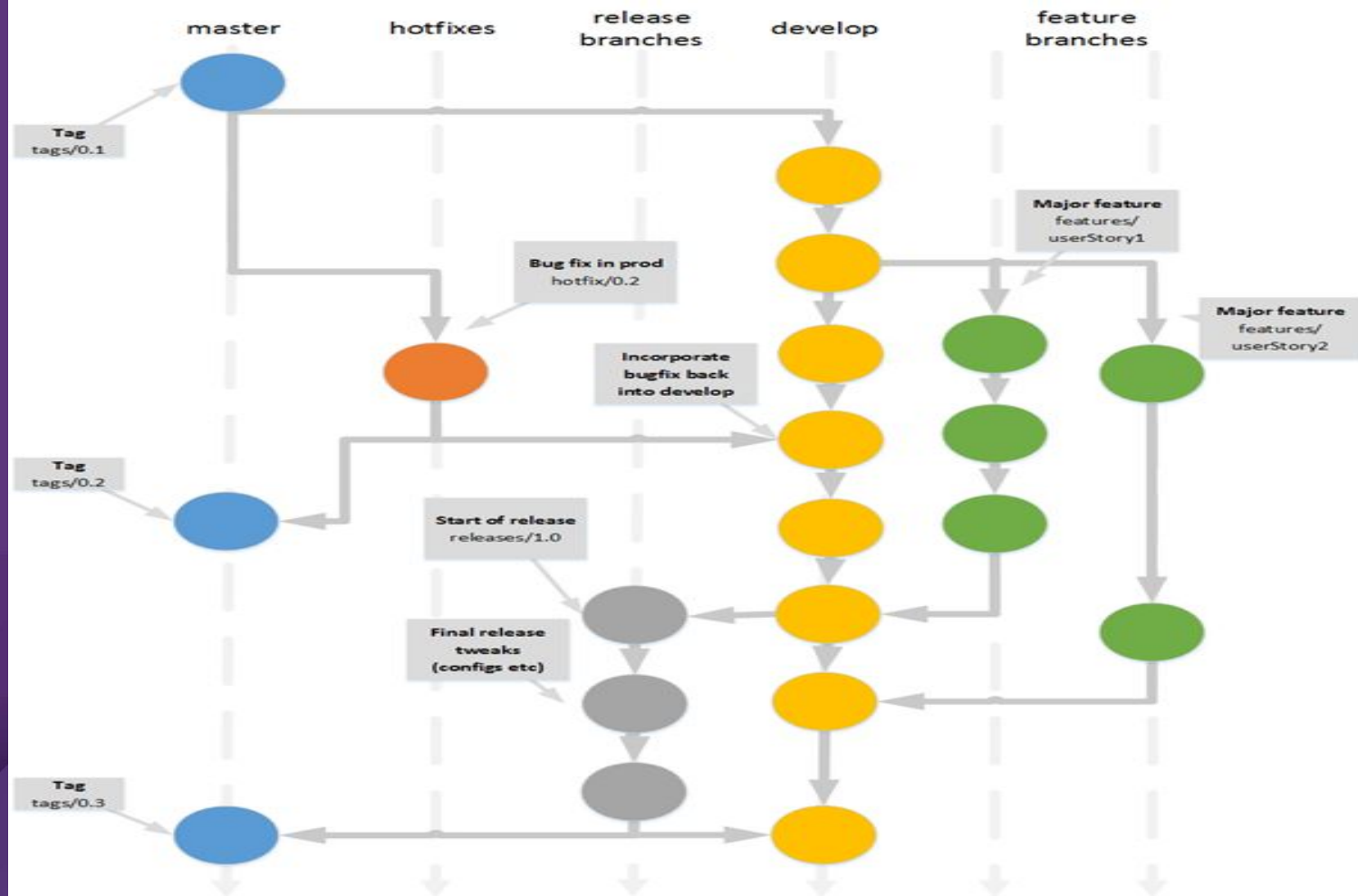
- ให้ลองเขียน สร้างไฟล์ .txt
- พิมพ์ข้อความอะไรก็ได้
- save และ ทำการ add file ขึ้น repository ของเรา
- เข้าไปเช็คไฟล์ของเราขึ้นไปหรือไม่ที่เว็บ git
- ลองลบไฟล์ .txt ออก
- ลอง git restore จาก repo มาที่เครื่องเรา

```
git add .  
git commit -m "title" -m  
"description"  
git push
```

```
git restore .
```



git with team





git branch

Branch in git



Software Park Thailand
Software Park Thailand
</Code Camp>

branch (สาขา) สร้างขึ้นแบ่งการทำงานออกเป็น
ส่วนๆ เพื่อป้องกัน และ ดัก code ที่มีปัญหาก่อนการ
deploy (เผยแพร่)



การแตก branch และ การสลับ branch

`git branch --list` : เป็นการเช็ค branch ว่ามี branch อะไรบ้าง

`git branch branch_name` : เป็นการแตก branch ใหม่

`git checkout branch` : เป็นการสลับ branch ในการทำงาน

`git checkout -b branch_name` : เป็นการสร้าง branch ใหม่ และ สลับไป branch พร้อมกัน



Software Park Thailand
</Code Camp>

Lab

branch

- สร้าง branch ใหม่ด้วย git branch
- checkout ไป branch ใหม่
- สร้าง ไฟล์ .txt
- แก้ไข ไฟล์ .txt
- upload file ขึ้น branch ใหม่
- สลับกลับมา master
- ดูการเปลี่ยนแปลงที่เกิดขึ้น

- ลบ branch ด้วย
- git branch -d branch_name
- ลอง ใช้คำสั่ง git checkout -b branch_name
- สร้างไฟล์ .txt upload
- สลับมา master
- ดูการเปลี่ยนแปลง

git command trick

```
git commit -am "title" -m "description"
```

```
git checkout -b name
```

```
git branch -m old_name new_name
```

```
git log --oneline --decorate --graph --all
```

```
git merge branch_name -m "description"
```

```
git merge branch_name --no-ff (no fast  
forward)
```

[มาลองเล่น git กัน](#)



git merge

การรวมข้อมูลระหว่าง branch

git merge เป็นการรวมข้อมูล จาก branch ไปอีก branch เช่น
เราต้องการ นำข้อมูล จาก develop ไปรวมกับ master

1. เราต้อง commit และ push ทุกอย่าง ของ develop ก่อน
2. สลับไปหน้า master git pull ให้เรียบร้อย
3. ใช้คำสั่ง git merge develop แก้ไข conflict

การรวมข้อมูลระหว่าง branch

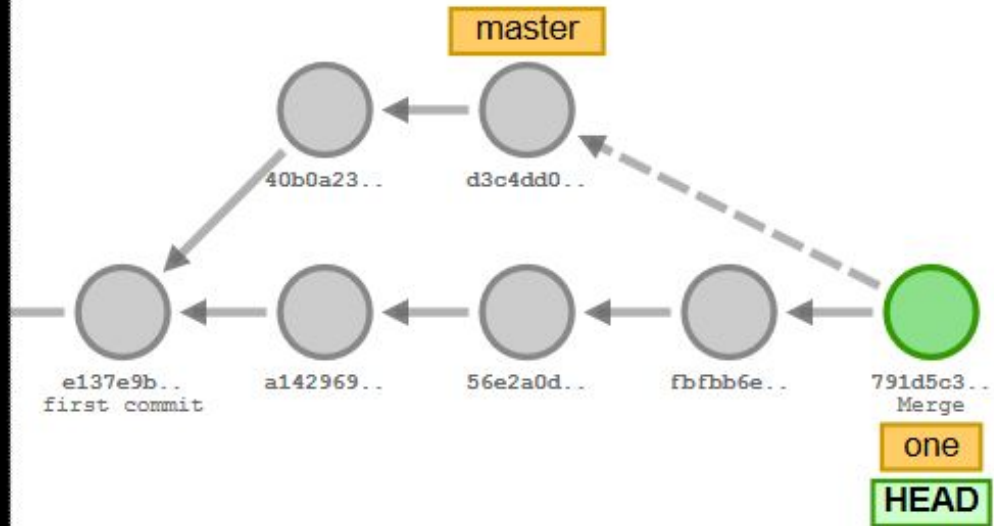
4. เมื่อแก้ไข conflict แล้ว ให้ save file นั้นๆ
5. git flow (git add => git push)
6. success รวมข้อมูลเรียบร้อยแล้ว

Have fun!

```
$ git branch one
$ git checkout one
$ git commit
$ git commit
$ git checkout master
$ git commit
$ git commit
$ git checkout one
$ git merge master
```

Local Repository

HEAD: one





Software Park Thailand
</Code Camp>

Lab

merge

กรณีมี conflict

- แก้ไข file .txt ใน master
- push ขึ้น repo
- สลับไป branch ที่สร้าง
เมื่อกี้
- แก้ไข file.txt ให้ไม่เหมือนใน
master
- push ขึ้น repo
- ทำการ merge เข้ากับ
master

กรณี ไม่มี conflict

- สลับมา branch ใหม่
 - แก้ไขไฟล์ .txt
 - push ขึ้น repo
 - ทำการ merge เข้ากับ master
อีกรอบ
- กรณีมี การเพิ่ม comment
- สลับมา branch ใหม่
 - แก้ไขไฟล์ .txt แบบ comment
 - push ขึ้น repo
 - ทำการ merge เข้ากับ master
 - อีกรอบ



Software Park Thailand
</Code Camp>





git rebase

git rebase

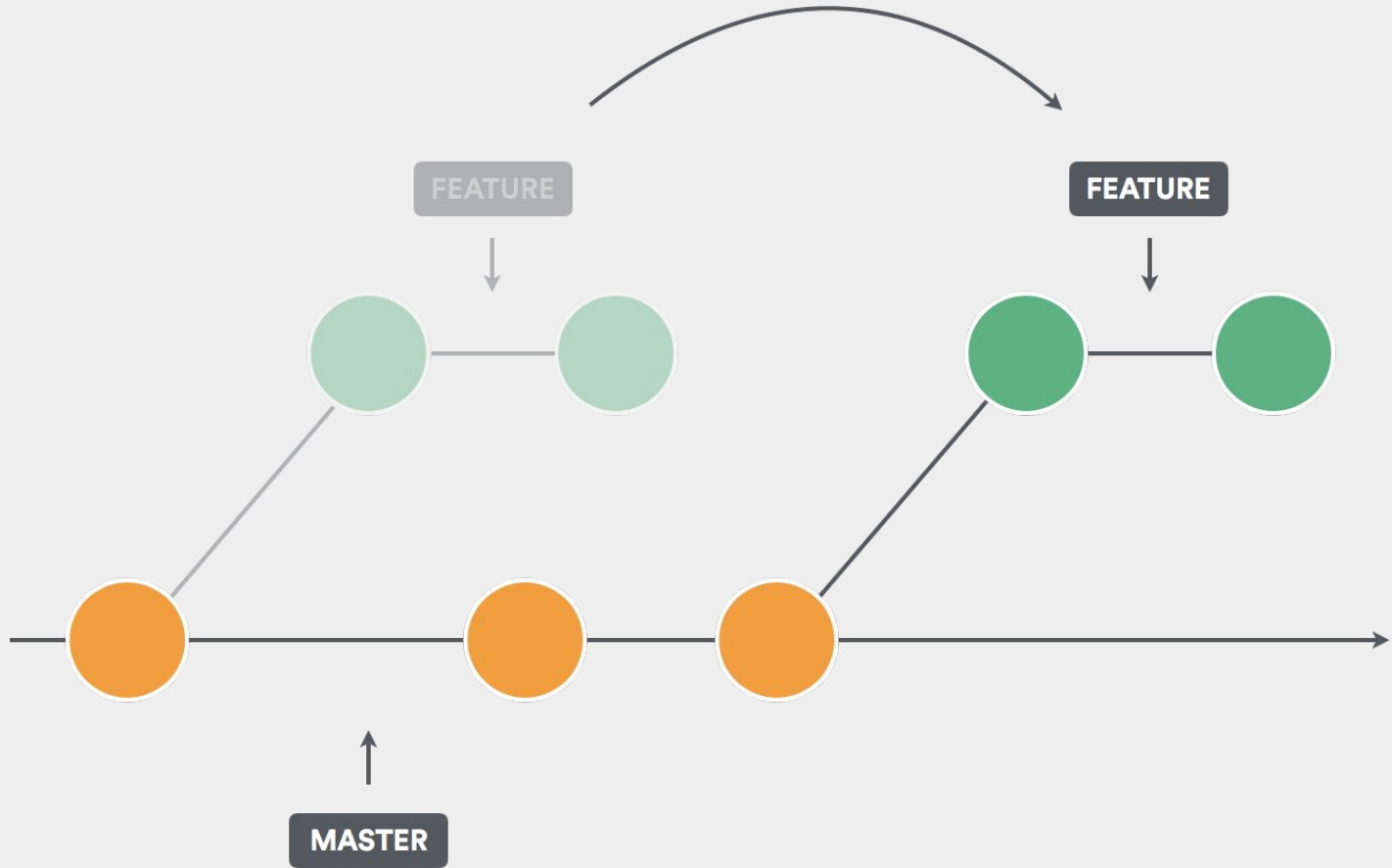
rebase เป็นการย้ายข้อมูล จาก branch หนึ่งมา
ต่ออีก branch

ขั้นตอนการทำ rebase จาก dev to master

1. update repo
2. checkout master
3. git pull
4. git rebase dev

git rebase

6. ตรวจสอบ conflict
7. ถ้ามีให้แก้ไข save แล้ว git add
8. ถ้าไม่มีให้ git add ได้เลย
9. git rebase --continue
10. git commit and push



Free Explore

Have fun!

```

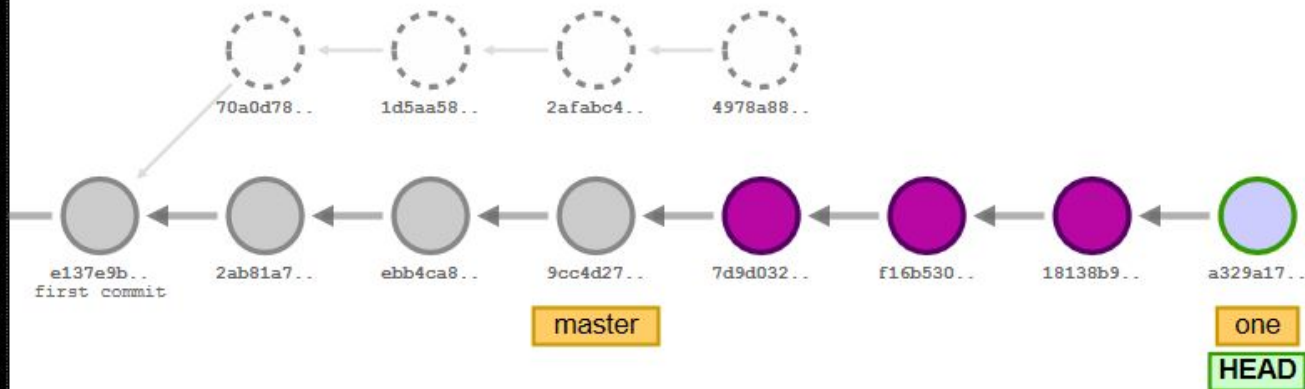
$ git checkout one
Cannot find commit: one

$ git branch one
$ git commit
$ git commit
$ git commit
$ git checkout master
$ git checkout one
$ git commit
$ git commit
$ git commit
$ git commit
$ git rebase master

```

Local Repository

HEAD: one





Software Park Thailand
</Code Camp>

Lab

rebase

กรณีมี conflict

- สร้าง branch ใหม่
- แก้ไข file .txt ใน master
- push ขึ้น repo
- สลับไป branch ที่สร้าง
เมื่อ
- แก้ไข file.txt ให้ไม่เหมือนใน
master
- push ขึ้น repo
- ทำการ rebase เข้ากับ
master



git ignore

git ignore

[ข้อมูลการใช้
เพิ่มเติม](#)

[การสร้าง gitignore](#)



git ignore เป็นการสร้างไฟล์ .gitignore ขึ้นมา เพื่อเป็นการกันไฟล์ที่ไม่ต้องการบางตัวขึ้น git repo ที่ เราได้ทำการ Remote ไว้

การเขียน gitignore

name/ : ทุก folder ที่มีชื่อนี้

/name/ : ทุก folder ที่มี name ในชื่อ

*.file : ทุกไฟล์ที่มีนามสกุลนี้

name.* : ไฟล์ที่ขึ้นต้นด้วยชื่อนี้ ใน path เดียวกัน

.file : ทุกไฟล์ที่มีนามสกุล .file และ .file อย่างอื่นด้วย

\! file.txt ! : ยกเลิกการ ignore ไฟล์ .txt ที่ขึ้นต้นด้วย file ทั้งหมด

**/ name : เฉพาะ file ชื่อ name หรือ folder ชื่อ name

a/** or a/**/b : ทุกไฟล์ใน folder a ทั้งหมด หรือ folder b ใน a

* = file

** = folder

! = ยกเลิก



Software Park Thailand
</Code Camp>

Lab

git ignore

ลอง สร้างไฟล์ .html, .css, .js
push repo

ignore ไฟล์ .html
git rm --cached file_name
push repo

edit ignore ไฟล์ .css
push repo
git rm --chached
file_name

